Duality in Computer Science

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
M. Sc. Silke Czarnetzki
aus Tübingen

Tübingen
2018

Tag der mündlichen Qualifikation:	30.01.2019
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Dr. Andreas Krebs
2. Berichterstatter:	Prof. Dr. Klaus-Jörn Lange

Zusammenfassung

Die Brücken zwischen Algebra, Logik und Topologie im Bezug auf Klassen regulärer Sprachen, sind in vielerlei Hinsicht erforscht worden: Das Blockprodukt verbindet Algebra und Logik, Strukturen aus der Logik lassen sich durch Transducer auf Sprachklassen abbilden, Sprachklassen wiederum sind durch Gleichungen zwischen Elementen eines topologischen Raumes beschreiben, welcher der Projektive Limes algebraischer Objekte ist. Dieser Auszug spiegelt nur die gröbsten der für die regulären Sprachen bekannten Zusammenhänge dar. Verlassen wir die regulären Sprachen, so betreten wir, was das Zusammenspiel zwischen Algebra, Logik und Topologie anbetrifft, größtenteils unbekanntes Terrain.

Diese Dissertation verfolgt das Ziel, die Beziehungen zwischen Algebra, Logik und Topologie auch außerhalb der regulären Sprachen besser zu verstehen. Grundlage hierfür bilden sogenannte Stone-Räume, die topologische Objekte für beliebige bool'sche Algebren stellen. In der Tat ist dies eine natürliche Erweiterung, da bereits die topologische Perspektive auf den regulären Sprachen auf Stone-Räumen arbeitet. Insbesondere lassen sich Konzepte, wie das syntaktische Monoid, auch als Stone-Räume konstruieren. Die Zusammenhänge, wie sich Algebra und Topologie in dieser Umgebung vereinen, werden zunächst erläutert.

Auf dieser Grundlage teilt sich die Arbeit in zwei Bereiche: Einen davon bilden die sogenannten visibly pushdown languages (VPL) – eine den regulären Sprachen in vielen Eigenschaften sehr verwandte Sprachklasse. Wir etablieren zunächst die Beziehungen zur Algebra: Hierfür werden adequate algebraische Erkennungsobjekte für VPL identifiziert, sogenannte Ext-Algebren. Im Gegensatz zu endlichen Monoiden, verfügen diese nicht ausschließlich über eine binäre Verknüpfung als Multiplikation, sondern über zusätzliche unäre Operationen, die die Struktur von VPL widerspiegeln. Im Folgenden wird ein Theorem, ähnlich zu dem von Eilenberg, bewiesen, welches eine eins-zu-eins Verbindung zwischen pseudo-Varietäten von VPL und pseudo-Varietäten von Ext-Algebren herstellt.

Für VPL wird die topologische Perspektive ähnlich etabliert, wie in der Konstruktion für die regulären Sprachen: Unter Verwendung von Ext-Algebren wird eine Metrik auf einer Teilmenge aller Wörter definiert. Die Vervollständigung des resultierenden metrischen Raumes ist dann der Stone-Raum der VPL, welcher ebenfalls eine Ext-Algebra bildet.

Über die Elemente dieses Raumes wird der Begriff der profiniten Gleichung für

Ext-Algebren gebildet. Über diesen Begriff wird es möglich eine Beziehung zwischen Mengen von Gleichungen und pseudo-Varietäten von Ext-Algebren herzustellen, welches die Eilnberg-Reiterman-ähnlichen Beziehungen zwischen Algebra und Topologie für VPL vervollständigt. Abschließend werden konkrete Gleichungen für eine spezifische pseudo-Varietät von VPL abgeleitet und in Zusammenhang zu den visibly counter languages (VCL) gesetzt.

Der zweite Bereich beschäftigt sich mit weitaus allgemeineren Klassen nicht-regulärer Sprachen, um eine Verbindung zu durch Logik definierten Sprachklassen herzustellen. Aufbauend auf dem Begriff der getypten Monoide gepaart mit sogenannten Stamps, wird der Begriff der getypten Stamps definiert, für welchen zunächst die Grundlagen etabliert werden, wie beispielsweise Spracherkennung und syntaktische getypte Stamps. Auch hierfür wird die algebraische Perspektive mit der topologischen durch eine Eilenberg-Reiterman-artige Verbindung verknüpft. Im Unterschied zu VPL kann diese Verbindung nicht mehr über die Vervollständigung eines metrischen Raumes hergestellt werden, sondern bedient sich projektiver Limiten. Es wird weiter gezeigt, dass die entstehenden topologischen Objekte wiederum enge Verbindungen zu Stone-Räumen aufweisen.

Diese Zusammenhänge lassen sich abschließend verwenden, um zunächst ein Blockprodukt auf getypten Stamps zu definieren, welches enge Verbindungen zu Logik aufweist, und über die zuvor entwickelten Konstruktionen ein Blockprodukt auf den topologischen Objekten zu definieren, welches die Zusammenhänge zu Logik erhält.

Abstract

The links between algebra, logic and topology have been examined in various ways on the case of the regular languages: The block product links algebra and logic, structures from logic are related to classes of languages via transducers, the classes of languages themselves are describable via equations on a topological space, who again is a projective limit of algebraic objects. This excerpt only mirrors the coarsest of the relations known for the regular languages. When we now leave the terrain of regularity concerning the interplay between algebra, logic and topology, we are entering largely unknown terrain.

This thesis intends to help in understanding the relations between algebra, logic and topology outside the regular languages. So-called Stone spaces, which provide topological objects for arbitrary Boolean algebras are the foundation of the investigation. Indeed, regarding Stone spaces provides a natural extension, since already in the case of the regular languages, the topological perspective works on special instances of Stone spaces. In particular, concepts like the syntactic monoid also are also derivable from Stone duality. We start by explaining how algebra and topology unite in this environment.

Based on that, the work is divided in two forks: One are the visibly pushdown languages (VPL) – a class that is in many ways still relatively close to the regular languages. We first establish the relationship to algebra by establishing adequate algebraic recognising objects, so called Ext-algebras. In contrast to finite monoids, those do not only have a binary operation that is the multiplication, but also an additional unary operation, which mirrors the structure of VPL. In the following, we prove an Eilenberg-like theorem, which establishes a one-to-one relationship between pseudo-varieties of VPL and pseudo-varieties of Ext-algebras.

For VPL, the topological perspective is established in quite a similar way as for the regular languages: Using Ext-algebras, we install a metric on a subset of all words. The completion of the resulting metric space then turns out to be the Stone space of the VPL, which again is an Ext-algebra.

On the elements of this space, we define the notion of profinite equations for Ext-algebras. The notion enables us to relate sets of equations and pseudo-varieties of Ext-algebras completing the Eilenberg-Reiterman-like triangle between algebra, topology and languages (VPL). Concluding, we derive equations for a concrete pseudo-variety of VPL and relate them to visibly counter languages.

The second fork is concerned with much more general classes of non-regular languages and tries to establish a relationship to classes of languages defined by logic. Building on the notion of typed monoids paired with so-called stamps, we define the notion of typed stamps, for which we first stablish the basics such as language recognition and syntactic typed stamps. Here also, we link the algebraic perspective with the topological one through an Eilenberg-Reiterman-like relationship. In contrast to VPL, the relationship cannot be obtained through completing a metric space, but builds on projective limits. We further show, that the resulting topological objects have close relationships to Stone spaces.

Lastly, those findings are used to define a block product first on typed stamps, which has close relations to logic and to subsequently extend it to a block product on the topological objects, preserving the relation to logic.

Preface

Acknowledgments

I would foremost like to thank my advisor Andreas Krebs, who got me interested into the subject in the first place. I am especially grateful for his incredibly quick intuition, which delivered fresh ideas and also enabled fluent high-level conversations about the research. Secondly, I would like to thank my co-advisor Klaus-Jörn Lange not only for the funding when I started, but also for his willingness to improve the accessibility of the papers, for the prompt help whenever necessary and for tea.

Many thanks go to Mai Gehrke, who also enabled my various stays in Paris, for the help in overcoming the barriers to entering the field of duality and for the fruitful discussions. The papers [CK16] and [BCGK17] are a result of this collaboration. In the course of my visits to Paris I also would like to thank Jean-Éric Pin for his insights from various years of working on related subjects and Howard Straubing for making the derived category theorem more accessible to me.

Next, I would like to thank all the people with whom I got to work with at the University of Tübingen, including but not limited to: Michael Ludwig, Demen Güler, Michaël Cadilhac, Charles Paperman, Sebastian Schöner, Ingo Skupin, Thomas Stüber, ... and those at the IRIF - especially Daniela Petrisan and Célia Borlido.

Last, but not least, I thank my family and friends for always having my back.

Chapter Notes

This thesis contains part of the research planned within the scope of the DFG-project "Duality in Circuit Complexity".

Chapter five contains results on topology and visibly pushdown languages, which can be found in [CKL18].

Chapters six and seven are based on [BCGK17], which in essence defines transductions and the block product on so-called Boolean spaces with internal monoids (BiMs), using an Eilenberg-like theorem and drawing the connection to logic. We diverge from the notions in there: Chapter four, for instance, contains a definition for the classes of recognisers, that deviates from the one in the paper. This results in two different Eilenberg-like theorems. The Reiterman-like theorem and the consideration

of equations are new and not contained in [BCGK17]. Thus, also chapter five is not concerned with BiMs, but with other profinite objects (we do not require closure under quotients of the Boolean algebras).

To my father

Immer strebe zum Ganzen,
und kannst du selber kein Ganzes werden,
als dienendes Glied
schließ an ein Ganzes dich an!

Johann Wolfgang von Goethe

Contents

List of Figures

1

Introduction

In order to solve a problem, it is important to be able to regard it from different angles, for difficulties arising from one point of view may be solvable from a second one and vice versa. It is thus that we consider the algebraic and topological treatment of non-regular languages and the relations between the two subjects.

The algebraic investigation of languages has its origin in the regular languages is often dated back to Kleene's theorem [Kle56], relating regular expressions and finite automata, which in their essence are algebraic objects. Another contribution which strengthened the algebraic approach was the *syntactic monoid*, which was defined 1956 by Schützenberger [Sch56], but is also credited to Myhill and Nerode, for instance in a paper by Rabin and Scott [RS59].

Those notions have led to a series of results on the regular languages.

Results Derived from Algebra

The notion of the syntactic monoid led, in 1964, to a decidable characterisation of the star-free languages by Schützenberger [Sch64]: A language is star-free if and only if its syntactic monoid has only trivial subgroups – a property that is verifiable by an algorithm and thus implies decidability. Alternatively, the equation $x^\omega = x^{\omega+1}$ characterises the same property and is read as follows: for each element x in the syntactic monoid of a star-free language, the idempotent x^ω it generates is equal to $x^\omega \cdot x$. This also gives an algorithm for decidability. It should not stay unmentioned that this class also has a logical characterisation by McNaughton and Papert [MP71]: they were able to show that the star-free languages coincide with the languages definable by first-order formulae with order-predicate. A few years earlier, Büchi [Büc60], had uncovered the equality of languages describable in monadic second order logic using the successor predicate and the regular languages.

A second instance, where the algebraic approach was put to use, are the quasi-

aperiodic languages. Quasi-aperiodicity is a property related rather to the syntactic morphism, than to the syntactic monoid: A morphism $\phi\colon A^* \to M$ into a finite monoid M is quasi-aperiodic if for each $n \in \mathbb{N}$, $\phi(A^n)$ contains no non-trivial groups. The languages recognised by those morphisms are precisely the regular languages contained in the complexity class $\mathbf{AC}^0$ [BCST92]. For instance the language PARITY does not have a quasi-aperiodic syntactic morphism and is thus not contained in $\mathbf{AC}^0$, as was shown by probabilistic means in the pioneering work of [FSS84]. The characterisation of the regular languages in $\mathbf{AC}^0$ in terms of morphisms was based mainly on two other results: One stating that $\mathbf{AC}^0$ corresponds to languages definable by first-order formulae with arbitrary predicates [Imm89, GL84] and one by Krohn and Rhodes [KR65], who provided a decomposition theorem for finite monoids, using the block product, which relates to the application of quantifiers in terms of logical formulae. In terms of logic, the regular languages in $\mathbf{AC}^0$ correspond to languages expressible by first-order formulae with regular predicates, which was an intermediate result of [BCST92].

The previous two classes are showcase examples for how the algebraic approach has improved our understanding of (subclasses of) the regular languages, but they are by far not the only ones. For instance, Simon's theorem [BS73] characterising the locally testable languages in terms of algebra is also counted amongst one of the considerably influential contributions.

An abstract characterisation of the relation between classes of finite monoids and classes of regular languages was found by Eilenberg in his study of pseudo-varieties [Eil76]. For instance, Schützenberger's characterisation of the star-free languages or Simon's theorem fall under that general relationship. The case of the quasi-aperiodic languages is a special one, since Eilenberg's study relates rather to monoids than to morphisms. The generalisation to so-called pseudo-varieties of stamps in [Str02] provides a generalisation of Eilenberg's theorem to classes of morphisms and classes of languages, which includes the quasi-aperiodic languages.

Since then, there have been many attempts to generalising these findings to even larger classes. An overview over the achievements of the algebraic approach on classes of regular languages can be found, for instance, in [RS97]. From the rich amount of results it is undeniable that algebra has been a central instrument in the investigation of regular languages, in particular those with a relation to fragments of logic – a relation that in its most general sense is covered by the block product. Explanations of the phenomenon can be found in Straubing's book [Str94] or the survey of Tesson and Thérien [TT07]. It is in particular to be held responsible for the many characterisations of regular languages inside circuit complexity classes, which for instance also led to the characterisation of the regular languages in $\mathbf{ACC}^0$ as the languages recognisable by solvable groups [BT88].

It is due to that history of results that the algebraic approach is being extended beyond the regular languages.

Leaving the Cage of Regularity

The notion of syntactic monoid also exists for non-regular languages, but it becomes quickly evident that the classic sense of recognition is not enough to maintain information about the languages, since the syntactic monoids of many non-regular languages contain a free monoid, which allows to encode arbitrary languages. Thus, concepts introducing additional structure were starting to be considered in the non-regular case: For instance in [Sak76] by Sakarovitch, who proposed to consider the syntactic monoid of a language and its syntactic image as a tuple.

A similar attempt to leave the regular languages was made in [KLR05], where an infinite monoid is equipped with not only one set as in the definition of Sakarovitch, but with a finite Boolean algebra of sets, the so-called typed monoids. There, the complexity class $\mathbf{TC}^0$ was characterised in terms of typed monoids through the definition of a block product on typed monoids. They were able to identify $\mathbf{TC}^0$ as the class of languages recognised by iterated block products of typed monoids relying on the integers equipped with the Boolean algebra generated by the positive integers. Similar to the characterisations of regular language classes, this proof was based on a relation between logic and the block product for typed monoids – a characterisation which was exploited in subsequent papers, to draw a link to majority logic with two variables and the smaller-predicate [BKR09] or more generally to two-variable logic and the block product in [BKM13].

Independently of the previously mentioned typed monoids, Gehrke, Griegorieff and Pin pursued a similar approach to contribute a notion of recognition through the addition of an ingredient previously successfully employed on the regular languages, namely topology. Before reviewing their work, we quickly summarise the impact of topology on the investigation of language classes.

A (Very) Brief History of Topology and Equations

An early contribution that drew a connection between topology and Eilenberg's pseudo-varieties of finite monoids, was made by Reiterman [Rei82] (who generalised a much earlier result by Birkhoff [Bir35]) in 1982. He discovered that each pseudo-variety of finite monoids is uniquely determined by so-called profinite equations. For instance, the class of all commutative finite monoids is determined by $xy = yx$ and the class of all aperiodic monoids by $x^\omega = x^{\omega+1}$, where this equation can be interpreted in exactly the same way as the same equation already described in the part on algebra. Formally, a profinite equation is an equality of two elements of the projective limit of all finite monoids – the free profinite monoid, whose elements are called profinite words. As such, a profinite equation defines a quotient of the free profinite monoid and each pseudo-variety of monoids has an associated quotient. Those profinite equations are particularly interesting, since in many cases, a characterisation through profinite equations already comes with a built-in algorithm for decidability, since equations, as in the case for $x^\omega = x^{\omega+1}$ can effectively be calculated on the syntactic monoid of a language. For instance, see the survey of Pin [Pin12].

Although Reiterman's theorem guarantees the existence of equations, it is not necessarily clear how to obtain them. This problem was approached by Almeida and Weil, using algebra – in particular the block product – in [AW95] where they gave concrete characterisations of the quotients of the free profinite monoids corresponding to pseudovarieties generated by block products of monoids. Later, through the derived category theorem [Til87] (resp. the kernel theorem [RT89]) they also obtained equations for pseudo-varieties built from block-products [AW98] and thus managed a first step towards a constructive approach to obtain equations: Given the defining equations of two pseudo-varieties $\mathbf{V}$ and $\mathbf{W}$, they were able to characterise the equations defining the block product of $\mathbf{V}$ and $\mathbf{W}$.

An observation made by Almeida [Alm95] and Pippenger [Pip97] was, that the free profinite monoid is just a special instance of a duality described by Stone [Sto36]. Stone uncovered that each Boolean algebra is isomorphic to a Boolean algebra of subsets of certain topological spaces, namely Stone spaces, and Almeida and Pippenger showed that the free profinite monoid was indeed the Stone space of the regular languages. In particular Stone duality was one of the main ingredients in the first paper by Gehrke, Grigorieff and Pin [GGP08], which via Stone duality drew a direct connection between classes of languages and equations, instead of doing the detour over finite monoids. In their paper from 2010 [GGP10], they used the fact that Stone duality is not limited to Boolean algebras of regular languages, introducing recognisers for arbitrary Boolean algebras of languages based on topology. In particular, they showed that these recognisers form a monoid if and only if the underlying Boolean algebra is regular and closed under quotients by words and that each Boolean algebra of languages also is determined by a set of so-called ultrafilter equations. A characterisation of a small class of non-regular languages trough ultrafilter equations was found in [GKP16] and the decidability of the regular languages therein was proven by translating the ultrafilter equations to profinite equations through a projection.

Research Uniting Algebra and Topology

On the regular languages, algebra and topology are undoubtedly related and the possibilities of translating back and forth between the perspective rather well explored. Outside the regular languages, this connection is not as well-established.

An attempt to obtain ultrafilter equations in a structured way through means of algebraic decomposition, as in [AW98], was made in [CK16]. This was done using a translation of a restricted version of the block product for typed monoids on the language side and resulted in equations for constant-size circuit classes. In [GPR16], the notion of Boolean space with internal monoids was established obtaining a closer relation between algebra and topology and the authors managed to derive equations for the Schützenberger product, which also can be thought of as a restricted version of the block product, that corresponds to the application of existential quantifiers on the logic side. Those results were generalised slightly more in [GPR17]. A full generalisation of the results of [AW95] was achieved in [BCGK17]. In merging the

notions of typed monoids, boolean spaces with internal monoids and stamps from [Str02], it was possible to define a notion of block product which relates to logic and to characterise its relation to the corresponding topological objects.

A second line of research relating algebra and topology, are forest algebras [BW08], that are finite algebraic objects for the recognition of trees. There, profinite objects based on finite forest algebras were considered in [Ali16].

It seems that regarding topological objects, such as Stone spaces, as a natural extension of finite algebraic objects (for instance finite monoids) might be the key to a better understanding of non-regular languages.

Structure and Results of the Thesis

The aim of this thesis is to strengthen the understanding of the interplay between algebra and topology on non-regular languages.

It begins with a short overview over notation, to keep everything documented. After that, we continue with the preliminaries, in which the necessary basics are reviewed, such as finite monoids, language recognition and logic on words. An extended review is dedicated to topology, since the concepts there are central to the thesis and readers not entirely familiar with them may hopefully profit from it.

The next chapter then aims at clarifying the connections between algebra and topology when concerned with languages: We first review the construction of the free profinite monoid and regard it afterwards from the perspective of projective limits of finite monoids. Leaving the regular languages, we continue with an explanation for Stone Duality and in particular ultrafilters, since these concepts are – without any intuition behind them – rather hard to grasp. Whenever it is adequate, we show where the algebraic perspective plays a role.

Having motivated the most important concepts, we start to extend them to classes of non-regular languages. The following chapter is concerned with the visibly pushdown languages, for which we first identify adequate finite algebraic recognisers, extending the findings of [AKMV05]. As a consequence, we obtain a syntactic object and an Eilenberg-like theorem, establishing a connection between classes of visibly pushdown languages and pseudo-varieties of finite algebraic objects. This lays out the necessary algebraic framework, which enables us then to build the topological framework on top of. In a similar way as for the regular languages, we define a metric over the finite recognisers and complete the resulting space. This provides us with a topological characterisation of the visibly pushdown languages, which we use to obtain a Reiterman-like theorem, which in turn gives us a notion of equations. The final section then uses this new notion of equations to give sound sets for two subclasses of the visibly pushdown languages.

The following chapter now leaves the approach of examining finite algebraic recognisers and rather uses possibly infinite algebraic objects equipped with a finite structure – similar to typed monoids [KLR05]. We establish the notion of typed stamps, which consist of a morphism, a possibly infinite monoid and a finite structure of sets on

that monoid. This particular design choice is made, since we are especially interested in recognisers for certain classes described by fragments of logic. Again, we first develop the necessary algebraic theory, such as syntactic objects and an Eilenberg-like theorem providing one-to-one relations between classes of typed stamps and classes of (arbitrary) non-regular languages. Out of these classes, we then build a projective system and characterise its projective limit, which now lies in a new category of recognisers – called dense stamps – generalising typed stamps. One essential ingredient to this generalisation are Stone spaces, which now replace the finite structure on typed stamps. We subsequently give a second representation of the dense stamp which is the projective limit and show that it may also be obtained purely from the corresponding class of languages. This in particular allows to switch points of view from the language side to the algebraic side and vice versa, whenever needed. We use it to develop a Reiterman-like theorem in which we show that each of the classes of typed stamps is determined by a set of ultrafilter equations.

In the last chapter, we use the topological and algebraic theory developed for typed stamps and dense stamps, in order to apply them to classes of languages defined by fragments of logic. For that purpose, we consider the principles of transduction and substitution (as in Tesson and Thérien [TT07]) and define an appropriate notion of transduction for typed stamps, which relates to substitution on the logic side. We then define the block product for typed stamps and via the transductions, prove that the block product may be used to model variable bindings by quantifiers on formulae describing languages. These findings are still restricted to single formulae and respectively single typed stamps, instead of classes. Consequently in the next section, we raise the notions of block product and substitution to classes of typed stamps and classes of formulae. By a slight detour over profinite alphabets, we are also able to show that the languages recognised by typed stamps in the block product of two classes of typed stamps may be encoded via only one transduction on a profinite alphabet, defined by a dense stamp. Finally, these notions allow us to define a block product on dense stamps and, in particular, on the projective limits of typed stamps – similar to Almeida and Weil for profinite monoids in [AW95] – and prove that if $\mathbf{V}$ and $\mathbf{W}$ are classes of typed stamps and $\mathbf{V}\Box\mathbf{W}$ is their block product, then the block product of the dense stamps corresponding to $\mathbf{V}$ and $\mathbf{W}$ via projective limits is (almost) equal to the dense stamp corresponding to $\mathbf{V}\Box\mathbf{W}$. This implies that also the block product on dense stamps is related to logic.

2

Notation and Terminology

We assume the reader is familiar with the following concepts and review the notation used throughout this document.

Sets

We denote by $\mathbb{N}$ the set of natural numbers including 0, by $\mathbb{Z}$ the set of integers and by $\mathbb{R}$ the set of real numbers.

We denote the empty set by $\emptyset$ and further: *union* ($\cup$), *intersection* ($\cap$), *complementation* (^{c}), *set difference* ($\setminus$), *subset* ($\subseteq$), *proper subset* ($\subsetneq$) and *membership* ($\in$).

Union, intersection and complementation are called *Boolean operations*.

If I is some set, then $(X_i)_{i\in I}$ is a *family of sets indexed by* I. *Direct products* of sets are written as $X \times Y$ or if they are indexed by some set as $\prod_{i\in I} X_i$ and their elements respectively are tuples $(x, y) \in X \times Y$ or $(x_i)_{i\in I} \in \prod_{i\in I} X_i$.

The *powerset* of a set X is denoted either by $\mathcal{P}(X)$ or by 2^X.

Relations

An *n-ary relation* on a set X is a subset of X^n. If a relation is 1-ary it is called a *unary* relation, 2-ary relations are called *binary*.

If R is a binary relation, we sometimes write it in infix notation, writing xRy instead of $(x, y) \in R$. A binary relation R is

- *reflexive* if xRx,
- *symmetrical* if xRy implies yRx,

- *transitive* if xRy and yRz implies xRz,
- *antisymmetric* if xRy and yRx implies $x = y$.

Moreover, R is an *equivalence relation*, if it is reflexive, symmetrical and transitive and R is a *partial order*, if R is reflexive, transitive and antisymmetric.

Functions

Let X, Y be sets. A function f from X to Y is denoted by $f\colon X \to Y$ and we use the notation $f(x) = y$ or $x \mapsto y$, meaning that f sends x to y. The set of all functions from X to Y is denoted by Y^X.

Let $f \in Y^X$. For $A \subseteq X$, the *image of A under f* is the set $f(A) := \{f(a) \mid a \in A\}$. The set $f(X)$ is called the *image of f*, X is called its *domain* and Y its *co-domain*. The function $g\colon A \to Y$ with $g(x) = f(x)$ is called the *restriction* of f to A, whereas the function $h\colon X \to f(X)$ with $h(x) = f(x)$ is called the *surjective co-restriction* of f.

Let $B \subseteq Y$, then $f^{-1}(B) := \{x \in X \mid f(x) \in B\}$. The set $f^{-1}(B)$ is called the *preimage* of B under f. The set $\{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$ is called the *kernel* of f.

Let Z be a set and $g \in Z^Y$. The *composition of f and g* is the function $g \circ f\colon X \to Z$ and defined by $(g \circ f)(x) = g(f(x))$. We say that *g factors through f*, if $f(x_1) = f(x_2)$ implies $g(x_1) = g(x_2)$ for all $x_1, x_2 \in X$. If g factors trough f, then there exists a function $h\colon Y \to Z$ such that $g = h \circ f$.

We say that f is *surjective* (*onto*) if for each $y \in Y$ there exists a $x \in X$ with $f(x) = y$, and that it is *injective* (*one-to-one*), if $f(x) = f(y)$ implies $x = y$.

If X is a set and $A \subseteq X$, we denote by $\chi_A\colon X \to \{0, 1\}$ the *characteristic function* of A, that is

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

The identity on X is denoted by $\mathrm{id}_X\colon X \to X$.

The map $\pi_X\colon X \times Y \to X$ with $\pi_X(x, y) = x$ is called the *projection* on X. If $\iota\colon Y \to X$ is an injective map, then ι is called an *embedding* and if in addition $Y \subseteq X$, then ι is called the *inclusion* of Y in X.

Comments on Terminology

Varieties: Observe that the term *variety* originally goes back to Birkhoff and references structures closed under arbitrary direct products. All classes in this document carrying the term variety in their name are either classes closed under finite direct product or classes related to them through recognition. Thus, all those

are *pseudo-varieties* in the sense of Eilenberg. We commonly use bold letters, like $\mathbf{V}$ for varieties of algebraic objects and italic letters, like $\mathcal{V}$, for the corresponding recognised classes of languages.

Equations: In the literature, one can find the terms equation and identity non-uniformly used for either the same term or to distinguish the two with an equation sometimes relating to closure under quotients by words. In this document, we use the term *equation* and do not assume any additional properties.

Circuit Classes: Whenever we refer to circuit classes, we refer to their non-uniform versions, that is: $\mathbf{AC}^0$ is the class of all constant depth, polynomial size circuits with Boolean gates, whereas $\mathbf{TC}^0$ is the class of all constant depth, polynomial size circuits with majority gates. In particular, $\mathbf{AC}^0$ corresponds to first-order logic with arbitrary predicates and $\mathbf{TC}^0$ to first-order logic with majority quantifiers and arbitrary predicates.

3

Preliminaries

While the following sections are not meant as a complete introduction to the subjects at hand and proofs are mainly omitted, we try to present the most important aspects in a coherent way, to keep this document self-contained. References to appropriate literature for background-reading are provided at the start of each section.

3.1 Algebra

An in depth-treatment of the connections between algebra and languages can be found in many books, we refer to Pin [Pin16] or Straubing [Str94]. For Boolean algebras, we refer to [Alm95].

Monoids

Recall that a monoid is a tuple $(M, \cdot)$, where M is a set, $\cdot\colon M \times M \to M$ a binary operation called the *multiplication* on M, which is associative and M has an identity element, denoted by 1 or 1_M, if it is not clear from the context, which satisfies $1 \cdot m = m \cdot 1 = m$ for each $m \in M$.

Unless it is necessary, we do not give the tuple and say that M is a monoid. The size of a monoid is equal to the size of the set M and denoted by $|M|$.

For $m \in M$, we define inductively $m^0 = 1$ and $m^k = m^{k-1} \cdot m$. The element m is called *idempotent*, if $m^2 = m$. Recall that in a finite monoid M, for each $m \in M$ there exists a $k \in \mathbb{N}$ such that m^k is idempotent. The element m^k is called the idempotent generated by m and denoted by m^ω. A monoid is called *aperiodic*, if for each $m \in M$, $m^\omega = m^\omega \cdot m$.

Moreover, if $n \cdot m = m \cdot n$ for all $n, m \in M$, then M is called *commutative*.

If M is a monoid, $N \subseteq M$ some subset and $m \in M$ an element of M, then we let

$$m^{-1}N = \{x \in M \mid m \cdot x \in N\} \text{ and } m^{-1}N = \{x \in M \mid x \cdot m \in N\}$$

If M is finite, we often illustrate the multiplication in a table. For instance

$\cdot$	0	1
0	0	0
1	0	1

reading $x \cdot y = T[x, y]$, where T is the table x marks the row and y the column. We denote the monoid above by U_1.

The monoid $\mathbb{Z}_k = \{0, \ldots, k-1\}$ is equipped with the multiplication $x \cdot y = x + y \mod k$.

Whenever not mentioned otherwise, we assume that both $\mathbb{N}$ and $\mathbb{Z}$ are equipped with the usual addition and thus form monoids.

A monoid M is called free on the generators $G \subseteq M$, if each element in M has a unique representation as a multiplication of elements of G. For instance $\mathbb{N}$ is the free monoid on the generators $\{1\}$.

We call a finite set of symbols A an alphabet and denote the free monoid on generators A by A^*. The elements of A^* are called words and represented as concatenations from A. The identity element of A^* is denoted by λ and called the empty word.

For any $w \in A^*$, we denote by $|w|$ the length of the word (that is the number of letters in w), where $|\lambda| = 0$. Moreover, for $a \in A$, we let $|w|_a$ be the number of as in w.

Morphisms

Let M, N be monoids. A morphism of monoids is a map $\varphi\colon M \to N$ satisfying $\varphi(x \cdot y) = \varphi(x) \cdot \varphi(y)$ for all $x, y \in M$.

If A is an alphabet, then each function $f\colon A \to M$ extends uniquely to a morphism $f^*\colon A^* \to M$ by letting $f^*(a) = f(a)$ for each $a \in A$ and inductively $f^*(x \cdot y) = f^*(x) \cdot f^*(y)$.

The monoid N is called a *quotient* of M, if it is the surjective morphic image of M, that is if there exists a morphism $\psi\colon M \to N$ such that N is the image of ψ. If $N \subseteq M$, then N is called a *submonoid* of M and finally, N *divides* M if N is the quotient of a sub- of M.

If the domain of a morphism is the free monoid A^*, we often define it solely on the generators of A^*. For instance, we let $h\colon A^* \to \mathbb{N}$ be the morphism sending any letter to 1 or just $(a \mapsto 1)_{a \in A}$ for the morphism mapping a word to its length.

Congruences

A congruence on a monoid M is an equivalence relation $\sim$ on M, which satisfies for all $x, y, m, n \in M$: If $m \sim n$, then also

$$xmy \sim xny.$$

The set of all equivalence classes is denoted by $M\backslash_\sim$. In particular, the multiplication $[m] \cdot [n] = [m \cdot n]$ on $M\backslash_\sim$ is well-defined and makes it a monoid with neutral element $[\lambda]$. The projection $\pi_\sim$ sending each element of M to its equivalence class with respect to $\sim$ is a monoid morphism. Hence each congruence defines a quotient of M.

Languages

A language over A^* is a subset $L \subseteq A^*$. We say that L is recognised by a monoid M if there exists a morphism $\varphi\colon A^* \to M$ such that $L = \varphi^{-1}(\varphi(L))$.

The syntactic congruence of a language $L \subseteq A^*$ is given by $u \sim_L v$ for $u, v \in A^*$ if and only if for all $x, y \in A^*$

$$xuy \in L \Leftrightarrow xvy \in L.$$

The syntactic monoid of L is $M(L) := A^*\backslash_\sim$ and recognises L via the syntactic morphism of L, denoted by η_L, which is the canonical projection.

3.1.1 Proposition Syntactic Monoid

A monoid M recognises a language L if and only if the syntactic monoid of L divides M.

Observe that the syntactic monoid of L recognises $v^{-1}L$ (resp. Lv^{-1}) for all $v \in A^*$. Whenever $\mathcal{C}$ is a class of languages and for each $v \in A^*$, $L \in \mathcal{C}$ implies $v^{-1}L \in \mathcal{C}$ and $Lv^{-1} \in \mathcal{C}$, then we say that $\mathcal{C}$ is closed under quotients by words.

The Block Product

The block product is a binary operation, which takes two monoids and generates a new one out of them. This construction is particularly useful for languages which are definable in some formalism of logic, since it is possible to model quantifier application with the block product. This relation is, apart from [Str94], also summarised in the survey of Tesson and Térien [TT07], which explains the relation of the block product to logic and subsequent results (on the regular languages).

Let M, N be monoids. We denote the multiplication on M by $+$ and on N by $\cdot$ (both need not be commutative, even if the notation with $+$ suggests so). The block

product of M and N, denoted by $M \square N$ is the set $(M^{N\times N} \times N)$ together with the multiplication

$$(f_1, n_1) \cdot (f_1, n_2) = (F, n_1 \cdot n_2),$$

where $f_1, f_2 \in M^{N\times N}$ and $n_1, n_2 \in N$ and where $F\colon N \times N \to M$ is the function given by

$$F(n_1', n_2') = f_1(n_1', n_2 n_2') + f_2(n_1' n_1, n_2')$$

for $n_1', n_2' \in N$.

Boolean Algebras

A Boolean algebra is a set $\mathcal{B}$ with two binary operations $\vee$ and $\wedge$ on $\mathcal{B}$ such that $(\mathcal{B}, \vee)$ (resp. $(\mathcal{B}, \wedge)$) forms a commutative monoid with neutral element 0 (resp. 1) and for all $x, y, z \in \mathcal{B}$

- $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ and $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ (distributivity)
- there exists an $\neg x \in \mathcal{B}$ such that $x \vee \neg x = 1$ and $x \wedge \neg x = 0$ (existence of complements).

For instance, if X is a set, then $\mathcal{P}(X)$ with $\vee = \cup$ and $\wedge = \cap$ is a Boolean algebra. Moreover, the set $\{0, 1\}$ with the operations

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

is called the two-element Boolean algebra.

Let $\mathcal{B}$ and $\mathcal{C}$ be Boolean algebras. A morphism of Boolean algebras is a map $f\colon \mathcal{B} \to \mathcal{C}$ such that for all $x, y \in \mathcal{B}$: $f(x \vee y) = f(x) \vee f(y)$ and $f(x \wedge y) = f(x) \wedge f(y)$. In particular, any morphism of Boolean algebras satisfies $f(0) = 0$, $f(1) = 1$ and $f(\neg x) = \neg f(x)$.

We say that $\mathcal{C}$ is a sub of $\mathcal{B}$ if $\mathcal{C} \subseteq \mathcal{B}$ and a quotient of $\mathcal{B}$ if there exists a surjective morphism $f\colon \mathcal{B} \to \mathcal{C}$.

3.2 Logic on Words

Logic on words is used to describe properties of words or languages by sentences in first and higher order logic. For instance the formula $\exists\, x\ Q_a(x)$ is true if and only if there is at least one position in a word containing an a and describes the language A^*aA^*. We consider only the case of first-order logic, since it is sufficient to understand the topics treated in this document. For a more in-depth presentation of the subject, we refer again to [Str94] or [Pin16].

We start by introducing syntax of first-order logic with informal semantics and continue to define the semantics on words formally. Formulas are built from variables, letter predicates, numerical predicates, quantifiers and Boolean connectives.

The symbols $\top$ and $\bot$ also are formulas and stand for the proposition that is always true ($\top$) or respectively, always false ($\bot$).

Variables

Positions in words are referenced by variables. We commonly use the symbols $x, y, z, x_0, x_1, \ldots, y_0, y_1, \ldots, z_0, z_1, \ldots$

Variables in formulas are called free unless they are bound by a quantifier (see below).

Letter predicates

For $a \in A$, we denote letter predicates by $Q_a(x)$ and the predicate is true if and only if the letter at position x is an a. Each letter predicate is a formula.

Numerical predicates

Numerical predicates are symbols with an arity $k \in \mathbb{N}$. For instance, the numerical predicate $<$ has arity 2. Moreover, each k-ary numerical predicate symbol is associated with a k-ary numerical relation, which in turn associates to each $n \in \mathbb{N}$ a subset of $\{1, \ldots, n\}^k$. For instance, the relation associated to $<$ sends each n to the set $\{(i, j) \mid i, j \in \{1, \ldots, n\} \text{ and } i < j\}$.

If R denotes a predicate symbol of arity k, then $R(x_1, \ldots, x_k)$ is a formula.

For the sake of readability, we cease to differentiate between the symbol and the numerical relation and assume that it is understood from the context. For instance let $n \in \mathbb{N}$. We write $(i_1, \ldots, i_k) \in R(n)$ meaning that the tuple $(i_1, \ldots, i_k)$ is contained in the set associated to n by R.

Quantifiers

Let ϕ be a formula with a free variable x and let $\mathcal{Q}$ be a symbol for a quantifier, then $\mathcal{Q}\, x\, \phi(x)$ is a formula and we say the variable x is bound by $\mathcal{Q}$. A formula with no free variables is called a sentence. For instance $\exists x\, Q_a(x)$ is a sentence.

Boolean connectives

As usual, we denote by $\wedge$ the logical `and`, by $\vee$ the logical `or` and by $\neg$ negation. If ϕ and ψ are formulae, so are $\phi \wedge \psi$, $\phi \vee \psi$ and $\neg\phi$.

Semantics

We already gave a sloppy interpretation of the semantics of formulae on words. For a precise definition, we make use of so-called $\mathcal{V}$-structures, which are defined inductively as follows.

Let A be a finite alphabet and let $n \in \mathbb{N}$. Moreover, for $i = 1, \ldots, n$ let $w_i \in A$.

1. The word $w = (w_1, \emptyset) \ldots (w_n, \emptyset)$ is a $\emptyset$-structure.

2. If $\mathcal{V}$ is a finite set of first-order variables, $w = (w_1, S_1) \dots (w_n, S_n)$ with $S_j \subseteq \mathcal{V}$ for $j = 1, \dots, n$ is a $\mathcal{V}$-structure and $x \notin \mathcal{V}$, then for each $i = 1, \dots, n$ the word

$$w_{x=i} := (w_1, S_1) \dots (w_i, S_i \cup \{x\}) \dots (w_n, S_n)$$

is a $(\mathcal{V} \cup \{x\})$-structure.

Observe that in a $\mathcal{V}$-structure as above, the set $\mathcal{V}$ is the disjoint union of the sets S_i. We denote by $A^* \otimes \mathcal{V}$ the set of all $\mathcal{V}$-structures over the alphabet A.

The semantics can now be defined inductively over the structure of formulae, if we identify each word $w \in A^*$ with a $\emptyset$-structure.

Let ϕ be a formula and let $\mathcal{V}$ be the set of free variables occurring in ϕ. In addition, let w be a $\mathcal{V}$-structure. Then we say that w models ϕ and write $w \models \phi$ if

- $\phi = Q_a(x)$ for some letter a, $w = u_{x=i}$ for some word $u \in A^*$ and $u_i = a$.
- $\phi = R(x_1, \dots, x_k)$ where R is a k-ary numerical predicate, $w = u_{x_1=i_1,\dots,x_k=i_k}$ for some word $u \in A^*$ and $(i_1, \dots, i_k) \in R(|w|)$.
- $\phi = \phi_1 \wedge \phi_2$ for some formulae ϕ_1 and ϕ_2 and both $w \models \phi_1$ and $w \models \phi_2$ hold.
- $\phi = \phi_1 \vee \phi_2$ for some formulae ϕ_1 and ϕ_2 and one of $w \models \phi_1$ or $w \models \phi_2$ holds.
- $\phi = \neg\psi$ for some formula ψ and w does not model ψ, also written $w \not\models \psi$.

To define the semantics of quantifiers on words, we use a simplification of monoidal quantifiers used in [BIS88] or respectively group-quantifiers in [KLR05].

Let ϕ be a formula with a free variable x and let w be a $\mathcal{V}$-structure, where $x \notin \mathcal{V}$, then we let

$$[w_{x=i} \models \phi] = \begin{cases} 1 & \text{if } w_{x=i} \models \phi \\ 0 & \text{otherwise.} \end{cases}$$

Moreover let M be a monoid, $T \subseteq M$ and $\delta\colon \{0,1\} \to M$ a map. Then we define the quantifier $\mathcal{Q}_{T,\delta}$ such that $w \models \mathcal{Q}_{T,\delta}\ x\ \phi(x)$ if and only if

$$\delta([w_{x=1} \models \phi]) \cdot \delta([w_{x=2} \models \phi]) \cdot \ldots \cdot \delta([w_{x=|w|} \models \phi]) \in T.$$

We define the following quantifiers:

- The existential quantifier $\exists$ stands for $\mathcal{Q}_{\{1\},\delta}$, where $\delta\colon \{0,1\} \to U_1$ maps 0 to 0 and 1 to 1.
- The for-all quantifier $\forall$ is given dually by $\forall\ x\ \phi(x) = \neg\exists\ x\ \neg\phi(x)$.
- The majority quantifier Maj stands for $\mathcal{Q}_{T,\delta}$ where $T = \{n \in \mathbb{Z} \mid n > 0\}$ and $\delta\colon \{0,1\} \to \mathbb{Z}$ maps 0 to -1 and 1 to 1.

The language defined by a formula ϕ over the alphabet A with free variables $\mathcal{V}$, is the set

$$L_\phi = \{w \in A^* \otimes \mathcal{V} \mid w \models \phi\}.$$

If $\mathbf{Q}$ is a set of quantifiers and $\mathcal{P}$ a set of numerical predicates, then $\mathbf{Q}[\mathcal{P}]$ denotes the set of all sentences built from quantifiers from $\mathbf{Q}$ and predicates from $\mathcal{P}$. For instance $\mathcal{N}$ denotes the set of all numerical predicates and FO the set of quantifiers $\{\exists, \forall\}$ hence FO$[\mathcal{N}]$ is the set of all formulae using $\exists$ and $\forall$ quantification and arbitrary numerical predicates. The set Maj$[\mathcal{N}]$ uses only Maj quantification. It was shown in [Lan04] that FO+Maj$[\mathcal{N}]$ = Maj$[\mathcal{N}]$.

3.3 Metric Spaces and Topology

Since topology is one of the key ingredients in this document, we dedicate an extended review to the basic notions. A condensed overview can also be found in [Pin16]. As for more detailed reading, there is a huge amount of books available on topology. In particular one by Eilenberg and Steenrod [ES52], which is probably one of the closest to the subjects of this document.

Metric Spaces

Metric spaces are a special instance of topological spaces, which will be treated later in that section. A metric space comes with a notion of distance, called a metric, that generalises the common notion of distance in three-dimensional space.

3.3.1 Definition Metric, Metric Space

Let X be a set. A metric on X is a map $d\colon X \times X \to [0, \infty)$ satisfying the properties that

1. d is positive definite: $\forall\, x, y \in X \;:\; d(x, y) = 0 \Leftrightarrow x = y$.
2. d is symmetric: $\forall\, x, y \in X \;:\; d(x, y) = d(y, x)$.
3. d satisfies the triangle inequality: $\forall\, x, y, z \in X \;:\; d(x, z) \leq d(x, y) + d(y, z)$.

The set X together with a metric is a metric space, denoted by (X, d).

If the metric on X is understood, we omit giving the tuple (X, d) and say that X is a metric space. Otherwise, if both X and Y are metric spaces, we commonly denote the metric on X by d_X and the metric on Y by d_Y, respectively.

In the rest of the subsection, we assume that X and Y always denote metric spaces.

Examples of metric spaces are:

- Any set X together with the *discrete metric* given by

$$d(x,y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{otherwise.} \end{cases}$$

- The real numbers $\mathbb{R}$ together with the map

$$d(x,y) = |x - y|.$$

- If $((X_i, d_i))_{i=1}^n$ is a (finite) family of metric spaces, then the product metric on $X_1 \times \cdots \times X_n$ is the metric given by

$$d(x,y) = \max_{i=1}^{n} \{d_i(x,y)\},$$

making $X_1 \times \cdots \times X_n$ a metric space.

Unless stated otherwise, finite spaces are always assumed to be equipped with the discrete metric and products always with the product metric.

Sequences are of special importance to metric spaces, in particular those that cluster around some point or in an area.

3.3.2 Definition Convergence

Let $(x_n)_{n\in\mathbb{N}}$ be a sequence of points in a metric space X. We say that the sequence $(x_n)_{n\in\mathbb{N}}$ converges to a point $x \in X$, if

$$\forall\, \epsilon > 0\ \exists\, n_0 \in \mathbb{N}\ \forall\, n \geq n_0\ :\ d(x_n, x) < \epsilon$$

We say that x is the limit of the sequence $(x_n)_{n\in\mathbb{N}}$ and write $\lim_{n\to\infty} x_n = x$.

Convergent sequences can be thought of as series of points that get arbitrarily close to some point in the space and it will often be convenient to describe a point in a metric space as the limit of some sequence.

3.3.3 Definition Cauchy sequence

A sequence $(x_n)_{n\in\mathbb{N}}$ in a metric space is a Cauchy sequence, if

$$\forall\, \epsilon > 0\ \exists\, N \in \mathbb{N}\ \forall\, n, m \geq N\ :\ d(x_n, x_m) < \epsilon.$$

Figure 3.1 displays a graphical representation of two Cauchy sequences. Intuitively speaking, Cauchy sequences are sequences whose points get arbitrarily close, clustering around some (possibly non-existent and thus more or less imaginary) point.

Observe that every convergent sequence is a Cauchy sequence, but the converse does not hold in general, as this example illustrates:

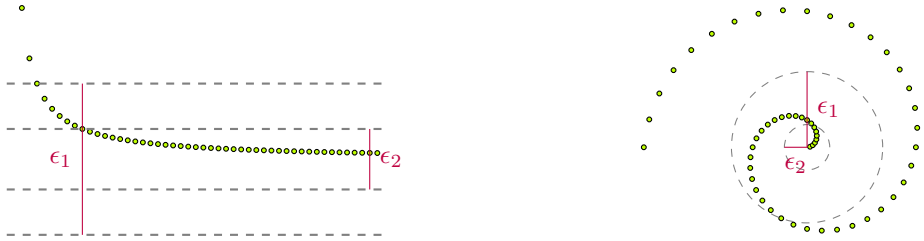


Figure 3.1: The illustration on the left shows the one-dimensional sequence $\frac{1}{n}$, where the ranges ϵ_1 (resp. ϵ_2) indicate the points that have distance ϵ_1 (resp. ϵ_2) to zero. The illustration on the right shows the sequence $(x \cdot \cos(x), x \cdot \sin(x))$, which is Cauchy with respect to the usual metric on $\mathbb{R}^2$.

If we consider the rational numbers $\mathbb{Q}$ together with the metric $d(x, y) = |x - y|$, then the sequence

$$\left(\left(1 + \frac{1}{n}\right)^n\right)_{n \in \mathbb{N}}$$

is Cauchy, but does not converge. It does converge in $\mathbb{R}$, however, to Euler's number e.

Metric spaces that have the property that every Cauchy sequence converges, are called *complete metric spaces*.

An example of a metric space that is not complete are, as the last example indicated, the rational numbers $\mathbb{Q}$, though the real numbers $\mathbb{R}$, are complete.

The Completion of a Metric Space

In fact, for each metric space X, there exists a complete metric space $\widehat{X}$ that "contains" X as a metric space. This space is called the completion of X. Before we can treat it in detail, we need some terminology.

Let X and Y be two metric spaces. An isometry is a map $\phi\colon X \to Y$, such that for all $u, v \in X$

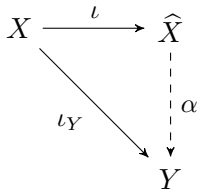
$$d_X(u, v) = d_Y(\phi(u), \phi(v))$$

Each isometry is automatically injective. Observe that if we are given an isometry from X to Y, this can be interpreted as X embedding into Y as a metric space. If an isometry is also surjective, we say it is an isometric isomorphism.

Let $D \subseteq X$. We say that D is dense in X, if for each $x \in X$, there exists a sequence $(x_n)_{n \in \mathbb{N}}$ in D that converges to x.

3.3.4 Proposition

Let X be a metric space. Then there exists a complete metric space $\widehat{X}$ and an isometry $\iota\colon X \to \widehat{X}$ such that $\iota(X)$ is dense in $\widehat{X}$.
The space $\widehat{X}$ is uniquely determined up to isometric isomorphism, that is: If Y is a complete metric space and $\iota_Y\colon X \to Y$ an isometry such that $\iota_Y(X)$ is dense in Y, then there exists a unique isometric isomorphism $\alpha\colon \widehat{X} \to Y$ such that the diagram



commutes. The space $\widehat{X}$ is a *completion* of X.

The property that makes $\widehat{X}$ uniquely determined up isometric isomorphism is also called the universal property of the completion.

Construction of the Completion

There is a canonical way to construct the completion, which we sketch below. In a sloppy way, we can say that every point of the completion is the limit of a Cauchy sequence in X. We do not include a full proof, but give a short overview over the basic ideas behind the construction, since some of them will be used in later chapters.

Let $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ be two Cauchy sequences in X. We say that $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ are equivalent, if the sequence

$$x_0, y_0, x_1, y_1, \dots$$

is also a Cauchy sequence and write $(x_n)_{n\in\mathbb{N}} \sim (y_n)_{n\in\mathbb{N}}$.

Observe that this is equivalent to the sequence $(d(x_n, y_n))_{n\in\mathbb{N}}$ converging to zero in $[0, \infty) \subseteq \mathbb{R}$.

We denote by $\mathrm{CF}(X)$ the set of all Cauchy sequences on X and define $\widehat{X}$ to be the set of all equivalence classes of $\mathrm{CF}(X)$ with respect to $\sim$. If $(x_n)_{n\in\mathbb{N}}$ is a Cauchy sequence, to avoid a mess of brackets, we denote by $[x_n]$ its equivalence class.

The map $\widehat{d}\colon \widehat{X} \times \widehat{X} \to [0, \infty)$ defined by

$$\widehat{d}([x_n], [y_n]) = \lim_{n\to\infty} d(x_n, y_n)$$

then is a metric on $\widehat{X}$.

Proof Sketch: It is not straight-forward that the limit on the right hand side of the definition exists. This is a consequence of some considerations on Cauchy sequences

and the fact that $\mathbb{R}$ is a complete space. That $\widehat{d}$ is independent from the choice of representatives follows directly from the triangle inequality of d and that $\widehat{d}$ is a metric from d being a metric.

Hence, $\widehat{X}$ together with the metric $\widehat{d}$ is a metric space.

3.3.5 Proposition

The space $\widehat{X}$ is complete.

Proof Sketch: Again, this is not trivial to show, but the general idea is to consider a Cauchy sequence $([x_n]_k)_{k\in\mathbb{N}}$ of equivalence classes of sequences which, without loss of generality, satisfies some properties we omit due to their technicality. Then, one can prove that the equivalence class of the sequence given by $y_j = x_{j_j}$ is indeed the limit of the above Cauchy sequence.

3.3.6 Proposition

Let $x \in X$ and $(x_n)_{n\in\mathbb{N}}$ be the constant Cauchy sequence given by $x_n = x$ for each $n \in \mathbb{N}$. Then the map

$$\begin{aligned} \phi\colon X &\to \widehat{X} \\ x &\mapsto [x_n] \end{aligned}$$

is an isometry and $\phi(X)$ is dense in $\widehat{X}$.

Since in this case, the proof is short, we give it explicitly.

That ϕ is an isometry follows from

$$\widehat{d}(\phi(x), \phi(y)) = \lim_{n\to\infty} d(x_n, y_n) = d(x, y).$$

To see that $\phi(X)$ is dense, pick $[x_n] \in \widehat{X}$ and $\epsilon > 0$. Then there exists an $n_0 \in \mathbb{N}$ such that for all $n, m \geq n_0$ we have $d(x_n, x_m) < \frac{\epsilon}{2}$ and hence

$$\widehat{d}(\phi(x_{n_0}), [x_n]) = \lim_{n\to\infty} d(x_{n_0}, x_n) \leq \frac{\epsilon}{2} < \epsilon$$

Hence, the given construction fulfills all requirements of Proposition 3.3.4 and is a completion of X.

In particular, any sequence that is ultimately equal to some element x, that is $x_n = x$ for all $n \geq n_0$ for some n_0, is in the same equivalence class as $\phi(x)$.

Continuity

Continuity is one important aspect of topology. The continuous functions are those that behave sufficiently well, to prove general statements, while most of these

statements are either wrong in general for non-continuous functions or extremely hard to prove.

3.3.7 Definition Continuity

A function $f\colon X \to Y$ between metric spaces is continuous, if for each convergent sequence $(x_n)_{n\in\mathbb{N}}$,

$$f\left(\lim_{n\to\infty} x_n\right) = \lim_{n\to\infty} f(x_n).$$

Hence continuous functions are precisely those that preserve limits.

An equivalent characterisation that is also often used as a definition for continuity, since it is easier to work with, yet less intuitive, is the following.

3.3.8 Proposition

A function $f\colon X \to Y$ between metric spaces is *continuous*, if and only if for all $x, y \in X$ and all $\epsilon > 0$, there exists a $\delta > 0$ such that

$$d(x,y) < \delta \Rightarrow d(f(x), f(y)) < \epsilon$$

A slightly stronger property is uniform continuity.

3.3.9 Definition Uniform continuity

A function $f\colon X \to Y$ between metric spaces is *uniformly continuous*, if for all $\epsilon > 0$, there exists a $\delta > 0$ such that for all $x, y \in X$

$$d(x,y) < \delta \Rightarrow d(f(x), f(y)) < \epsilon$$

It is clear that uniform continuity implies continuity. That the converse direction is generally wrong becomes evident, when we verify that the function $f\colon (0,\infty) \to (0,\infty)$ given by $f(x) = \frac{1}{x}$ is continuous but not uniformly continuous.

We will often use that $d(f(x), f(y)) \leq d(x,y)$ implies uniform continuity.

The following Corollary is a consequence of the universal property of the completion.

3.3.10 Corollary

If Y is a complete metric space and $f\colon X \to Y$ is a uniformly continuous function, then there exists a unique uniformly continuous function $\widehat{f}\colon \widehat{X} \to Y$ that extends f. In particular, if E and F are metric spaces, then any uniformly continuous function $g\colon E \to F$ extends to a unique uniformly continuous function $\widehat{g}\colon \widehat{E} \to \widehat{F}$.

Thus, it is for complete metric spaces often natural to consider uniformly continuous functions.

Metric Topology

This subsection is primarily to be seen as a motivation and guideline for the following part on general topology and to illustrate how general topology is an extension of metric spaces. Some of the properties here will be found later on as definitions for the same notions extended to topological spaces.

Let $r > 0$ and $x \in X$, then the set

$$B_r(x) = \{y \in X \mid d(x, y) < r\}$$

is called the open ball of radius r centered at x.

A subset $U \subseteq X$ is called *open*, if for each $x \in U$, there exists an $r > 0$ such that

$$B_r(x) \subseteq U.$$

With these definitions, one can show that each open ball of radius r is an open set and the open sets of X are precisely the unions of open balls.

3.3.11 Proposition

Let Ω be the set of all open sets of X, then

1. $\emptyset, X \in \Omega$
2. if $U, V \in \Omega$, then so is $U \cap V$
3. if I is some index set and $A_i \in \Omega$ for each $i \in I$, then the union of all A_i is also contained in Ω.

To summarise, the open sets of a metric space are closed under finite intersections and arbitrary unions.

Also continuous functions are determined by their behaviour on open sets:

3.3.12 Proposition

A function between metric spaces $f\colon X \to Y$ is continuous if and only if preimages of open sets are open, that is: If U is open in Y, then $f^{-1}(U)$ is open in X.

General Topology

We will try to link the definitions from general topology with those of metric topology to highlight where they connect and to provide examples.

3.3.13 Definition Topology, Topological Space

Let X be a set. A *topology* Ω on X is a subset of $\mathcal{P}(X)$ whose elements are called *open sets* and which

1. contains $\emptyset$ and X,
2. is closed under finite intersections and
3. is closed under arbitrary unions.

The set X together with Ω is called a topological space and written (X, Ω).

We assume from now on that X is a topological space and omit mentioning the topology Ω explicitly, whenever it is not necessary.

Examples of topologies are:

- The *discrete topology*, which is the full powerset of X. That is, every set is open.
- The *trivial topology*, in which only $\emptyset$ and X are open.
- The *co-finite topology* which consists of all sets that are either the empty set or have finite complement.

Observe that the axioms of a topology are satisfied by the open sets of a metric space (see Proposition 3.3.11) and hence every metric induces a topology, which contains precisely the unions of open balls.

We say that a topology Ω on X is generated by some family of subsets $\mathcal{G} \subseteq \mathcal{P}(X)$ if Ω is the smallest topology containing $\mathcal{G}$.

A set $\mathbf{B}$ of subsets of X with the property that each element of Ω is a union of arbitrarily many elements of $\mathbf{B}$ is called a *basis* for the topology Ω, if

Hence, the open balls of a metric space form a basis of the topology induced by the metric.

We consider topologies on products and subsets:

Let $(X_i)_{i \in I}$ be a family of topological spaces. Then their product $\prod_{i \in I} X_i$ is equipped with the topology generated by the sets $\prod_{i \in I} K_i$, where $F \subseteq I$ is a finite set and K_i is some open set in X_i, if $i \in F$ and otherwise K_i equals X_i. Observe that these sets form a basis. This topology is called the *product topology*.

If $(X_i)_{i \in I}$ is a finite family of metric spaces, then the topology induced by the product metric is precisely the product topology of the topologies on the factors.

If X is a topological space and $Y \subseteq X$ a subset, then Y is equipped with the *subset topology*, which is generated by all sets $K \cap Y$, where K is open in X.

Important Terms

We list some further terms that are necessary for the topics covered later on:

3.3.14 Definition Closed and Clopen Sets

A set subset C of a topological space is called *closed*, if its complement in X is open. If both C and its complement are closed (respectively open), C is called *clopen*.

For instance, if X is a metric space, $r > 0$ and $x \in X$, then the set

$$\overline{B}_r(x) = \{y \in X \mid d(x, y) \leq r\}$$

is closed.

In general, if A is some subset of a topological space, we denote by $\overline{A}$ the smallest closed set containing A – which exists since closed sets are closed under arbitrary intersections – and call it the *topological closure* of A.

It is not too hard to see that, $\overline{B}_r(x)$ contains for each converging sequence in $\overline{B}_r(x)$ also its limit point. This in fact holds for every closed set of a metric space: A subset C of a metric space X is closed if and only if for every converging subsequence $(x_n)_{n\in\mathbb{N}}$ in C, the limit of $(x_n)_{n\in\mathbb{N}}$ is also contained in C.

Generalising the term dense subset from metric spaces, we say that a subset D of a topological space X is dense in X if and only if $\overline{D} = X$.

3.3.15 Definition Zero-Dimensional

A topological space is called *zero-dimensional* if it has a basis of clopens, that is each open set is a union of clopen sets.

For instance, any discrete space is zero-dimensional.

3.3.16 Definition Hausdorff

A topological space X is called *hausdorff*, if for each two points $x, y \in X$, there exist disjoint open sets O_x and O_y such that $x \in O_x$ and $y \in O_y$.

It is not too hard to see that any two points x, y of a metric space can be separated by open balls of radius $\frac{d(x,y)}{2}$ and hence every metric space X is hausdorff.

A topology that is not hausdorff is the co-finite topology on $\mathbb{N}$, since no two sets in the topology have non-empty intersection.

Similar to the notion of topology itself, where the properties of open sets in a metric space form the definition of a topology, what was a conclusion about continuous functions in metric spaces is now definition for topological spaces.

3.3.17 Definition Continuity

A function $f\colon X \to Y$ between topological spaces is called continuous, if preimages of open sets are open.

A fact that we will frequently use is the following:

3.3.18 Proposition

If X and Y are topological spaces, Y is hausdorff and D is a dense subset of X, then any continuous function $f\colon D \to Y$ has a unique continuous extension $f\colon X \to Y$.

The next definition is to topological spaces, what continuity is to functions: Compact spaces provide nice-to-have properties, which simplify proofs and which non-compact spaces do not have in general. It is also often argued that compact spaces are "the next best thing to finite spaces", since many propositions about functions on finite spaces are true for continuous functions on compact spaces, such as: A continuous function into a compact space has a minimum and maximum.

3.3.19 Definition Compact

A subset K of a topological space X is called *compact*, if for each collection of open sets $(O_i)_{i\in I}$ with $K \subseteq \cup_{i\in I} O_i$, there exists a finite set $F \subseteq I$ such that $K \subseteq \cup_{i\in F} O_i$.

For instance, each finite space with the discrete topology is compact and any infinite space with the discrete topology is not compact.

One beautiful aspect also is the possibility to translate from "local" to "global" properties. If a property of sets is preserved by finite unions, then it suffices to show that each point in the space has some open neighbourhood with that property, to show that the whole space has the property.

3.4 Projective Limits

Projective limits occur, for instance, in algebraic number theory: The p-adic numbers are a projective limit of finite groups. In later chapters, we are also going to investigate algebraic properties through projective limits. For background reading, we refer to [ES52] or [Alm95]. In order to define the notion properly, we need some category-theoretic terminology.

Categories

Recall that a category $\mathcal{C}$ consists of a class of objects $\mathrm{Obj}(\mathcal{C})$ and a class of morphisms $\mathrm{Hom}(\mathcal{C})$, which are maps between objects of $\mathcal{C}$. Each morphism has a domain and a

co-domain in $\mathrm{Obj}(\mathcal{C})$ and we write $f\colon X \to Y$ for a morphism with domain X and co-domain Y.

Moreover, $\mathcal{C}$ admits for a composition of morphisms $\circ$, such that if $f\colon X \to Y$ and $g\colon Y \to Z$ are morphisms, then $g \circ f\colon X \to Z$ is a morphism. Additionally, $\circ$ is associative and for each object X, there exists an identity morphism $1_X\colon X \to X$ which behaves like a neutral element with respect to composition of morphisms.

Partially ordered sets

Recall that a partial order over a set I is a binary relation $\leq$ on I, that is reflexive, antisymmetric and transitive. A set I together with a partial order is called partially ordered set (or poset) . A poset I is *directed*, if any two elements of I have a common upper bound, that is for all $i, j \in I$ there exists an $s \in I$ such that $i \leq s$ and $j \leq s$.

For instance, the set of all finite monoids with division is a directed poset and a common upper bound of two finite monoids is their product.

Projective Systems and Limits

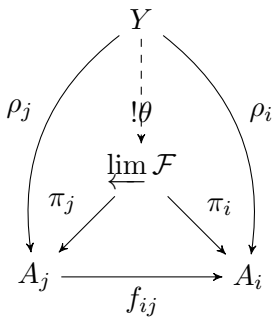
Let $\mathcal{C}$ be a category, I a directed poset and let $(A_i)_{i\in I}$ be a family of objects in $\mathcal{C}$ indexed by I. For each ordered pair $i \leq j$, let $f_{ij}\colon A_j \to A_i$ be a morphism in $\mathcal{C}$. The collection $(A_i)_{i\in I}$ together with the morphisms f_{ij} is called a *projective system*, if it satisfies the following properties:

- For each $i \in I$, the morphism $f_{ii}\colon A_i \to A_i$ is the identity on A_i.
- For all $i, j, k \in I$ with $i \leq j \leq k$, we have $f_{ik} = f_{ij} \circ f_{jk}$.

Since it is often the case that the family of objects and the directed poset I are identical, we often say that some directed poset forms a projective system.

Projective systems are also known under the name of inverse limit systems, inverse systems or cofiltered diagrams. The *projective limit* (also inverse limit or cofiltered limit) of a projective system $\mathcal{F}$ with objects $(A_i)_{i\in I}$ and morphisms f_{ij}, is denoted by $\varprojlim \mathcal{F}$ and comes with maps $\pi_i\colon \varprojlim \mathcal{F} \to A_i$ that are compatible with the morphisms f_{ij}, that is $\pi_j = f_{ji} \circ \pi_i$. It is defined by the following universal property:

For each Y with morphisms $(\rho)_{i\in I}$ and $\rho_i\colon Y \to A_i$ there exists a unique morphism θ, such that the diagram



commutes. If $\mathcal{C}$ is a concrete category, that is a category in which the objects are essentially sets with additional structure, and in which the infinite product exists, the inverse limit is the subset (respectively subalgebra) of the product of all A_i, given by

$$\varprojlim \mathcal{F} = \left\{ (x_i)_{i\in I} \in \prod_{i\in I} A_i \;\middle|\; \forall\ i \leq j,\ f_{ij}(x_j) = x_i \right\}.$$

Observe that the projective limit of a projective system in some category is not necessarily an object in the same category. For instance, the projective limit of the finite monoids

$$S_i = \{1, a, \ldots, a^i\}$$

with the multiplication $a^n \cdot a^m = a^{\min\{n+m,i\}}$ and connection morphisms

$$\begin{aligned} f_{ij} \colon S_j &\to S_i \\ a^n &\mapsto a^{\min\{n,i\}} \end{aligned}$$

is the set $\{a\}^* \cup \{0\}$, which is not a finite monoid.

If $\mathcal{F}$ is a projective system of topological spaces $(A_i)_{i\in I}$, then the projective limit, as a subspace of the product $\prod_{i\in I} A_i$ is equipped with the subspace topology, whereas the product is as usual equipped with the product topology.

4

Algebra Meets Topology

This chapter is meant as a review on how both algebra and topology are known to interact on the regular languages and where these results fit into a larger framework, namely Stone Duality.

4.1 A Review of the Regular Languages

Probably *the* prime example where algebra and topology are tied together is the free profinite monoid. We are going to present two possible constructions of the free profinite monoid: as the completion of a metric and as the projective limit of finite monoids. For more in-depth reading on the first, we refer to [Pin16] and on the second to [Alm95].

Recall that the free profinite monoid is the completion of A^* with respect to a metric that intrinsically relies on algebra, that is:

A finite monoid M *separates* two words $u, v \in A^*$, if there exists a morphism $\varphi\colon A^* \to M$ such that $\varphi(u) \neq \varphi(v)$. Then, we define

$$r(u,v) = \min\{|M| \mid M \text{ is a finite monoid separating } u \text{ and } v\}$$

and the metric on A^* is given by $d(u,v) = 2^{-r(u,v)}$.

That the topology induced by that metric also keeps algebraic information is illustrated in Figure 4.1: Words that cannot be separated by a monoid of size 2 – and thus have distance less than $\frac{1}{4}$ – are indicated in the same color. Since the only monoids of size two are U_1 and $\mathbb{Z}_2$, one can observe the change in parity when exchanging one letter, that $\mathbb{Z}_2$ contributes. The two words that are separated by U_1 are the ones consisting of all as or all bs.

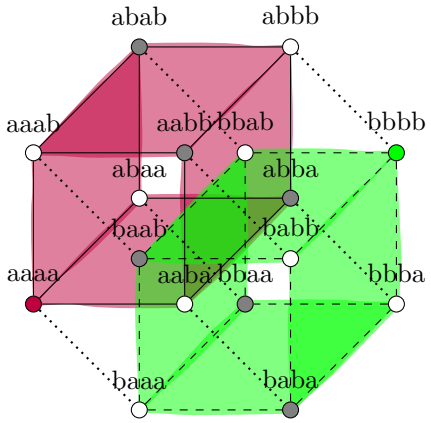


Figure 4.1: Words of length four that cannot be separated by a monoid of size 2 and thus have a distance smaller than 2^{-2} are indicated in the same color.

Recall the topology induced by the metric d is discrete, since if we let $u \in A^*$ and $n \in \mathbb{N}$ be the size of the syntactic monoid of $\{u\}$, then the open ball of radius 2^{-n} around u is equal to $\{u\}$.

Its completion, the free profinite monoid, however, is not discrete. Elements of the free profinite monoid also carry algebraic information. For instance, for $x \in A^*$, the limit of the sequence $x^{n!}$,

$$x^\omega = \lim_{n\to\infty} x^{n!}$$

can be thought of as an idempotent in a finite monoid for the following reason: Each map $\varphi\colon A \to M$ into a finite monoid M extends uniquely to a continuous monoid morphism $\widehat{\varphi}\colon \widehat{A^*} \to M$, which maps the limit of $x^{n!}$ to the limit of the sequence $\widehat{\varphi}(x)^{n!}$ in M. Since this sequence is ultimately equal to some idempotent e of M and M is discrete, the limit of the sequence is also equal to e.

For instance, if

$$\begin{aligned} \varphi\colon A^* &\to \mathbb{Z}_2 \\ w &\mapsto |w| \mod 2 \end{aligned}$$

then $\widehat{\varphi}(x^\omega) = 0$ for any $x \in A^*$, since 0 is the only idempotent in $\mathbb{Z}_2$.

If m is an element of an aperiodic monoid and m^ω is the idempotent generated by m, then $m^\omega \cdot m = m^\omega$. This intuitively explains the characterisation of the aperiodic monoids through the equality

$$x^\omega = x^{\omega+1}$$

where $x^{\omega+1} := x \cdot x^\omega = x^\omega \cdot x$.

Beside the characterisation of the free profinite monoid as the completion of a metric space, there is a second and less frequently mentioned one. This connection does not only explain the name of the object, but also opens up a second perspective,

why topology and algebra interact so closely on it: The free profinite monoid is the projective limit of a projective system over finite monoids.

Projective Limits and Topology

We review the construction of the free profinite monoid briefly. In order to build the projective system for the free profinite monoid, we consider so-called A-generated monoids.

An A-generated monoid is a map $\mu\colon A \to M$ into a monoid M such that $\mu^*\colon A^* \to M$ is surjective. A morphism from $\mu\colon A \to M$ to the A-generated monoid $\nu\colon A \to N$ is a (unique) map $\psi\colon M \to N$, such that $(\psi \circ \mu) = \nu$. Observe that this induces a partial order on the set of all A-generated monoids by letting $\nu \leq \mu$ if and only if there exists a morphism of A-generated monoids from μ to ν.

We let I be the poset of all finite A-generated monoids, that is those, where the co-domain of the maps is a finite monoid. Observe that I is a directed poset, where the common upper bound of μ and ν is the product map sending w to $(\mu(w), \nu(w))$.

To each $\mu \in I$, we associate a finite monoid M_μ that is the co-domain of μ.

The collection $(M_\mu)_{\mu \in I}$ forms a projective system, where for any $\nu \leq \mu$, the connection morphism $f_{\nu\mu}\colon M_\mu \to M_\nu$ is equal to the morphism of monoids induced by the unique morphism of A-generated monoids ψ from μ to ν.

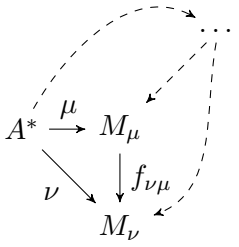


Figure 4.2: Part of the projective system visualised, where ... may be replaced by any finite A-generated monoid larger than M_μ and M_ν or by the projective limit.

In the category of (possibly infinite) monoids the projective limit of the projective system $(M_\mu)_{\mu \in I}$ is the submonoid of the direct product of all M_μ

$$X := \varprojlim_{\mu \in I} M_\mu = \left\{ (x_\mu)_{\mu \in I} \in \prod_{\mu \in I} M_\mu \;\middle|\; \forall \nu, \mu \in I \text{ with } \nu \leq \mu \;:\; f_{\nu\mu}(x_\mu) = x_\nu \right\}.$$

The monoid X contains A^* via the morphism ι sending x to $(\mu(x))_{\mu \in I}$, which is injective, but not surjective: For instance, for any $x \in A^*$ let x^ω be the element of X such that the projection $\pi_\mu(x^\omega)$ equals the unique idempotent generated by $\mu(x)$. This element is not in the image of A^*, since for any $x \in A^+$, there exists a finite A-generated monoid μ, such that $\mu(x)$ is not idempotent.

From an intuitive point of view, the definition of x^ω on X is somewhat compatible with the topological one, as both relate to idempotents. This is no coincidence:

Recall that each finite monoid also may be seen as a discrete topological space and X has the subspace topology of the product topology. Then, the sequence $\iota(x^{n!})$ converges to $x^\omega \in X$.

It also follows that the projections $\pi_\mu \colon X \to M_\mu$ are continuous, and hence any set of the form $\pi_\mu^{-1}(K)$ with $K \subseteq M_\mu$ is clopen. In particular on may prove that

$$\pi_\mu^{-1}(K) = \overline{\iota(\mu^{*-1}(K))},$$

since A^* is dense in X. This observation serves as an intuitive explanation, that the closures of languages recognised by finite monoids are precisely the clopen sets:

4.1.1 Proposition

Let $L \subseteq A^*$, then L is regular if and only if $\overline{L} \subseteq X$ is clopen, where $\overline{L}$ identifies the closure of $\iota(L)$.

The observation made by Pippenger [Pip97] and Almeida [Alm95] places this fact about the clopens of the free profinite monoid in a wider context: The free profinite monoid is the Stone space of the regular languages.

Stone spaces are compact, hausdorff and zero-dimensional spaces – properties one may verify the free profinite monoid satisfies. In particular, each Boolean algebra has an associated Stone space.

Since Stone spaces exist for arbitrary Boolean algebras, not only regular ones, they may well be a key to the better understanding of classes of non-regular languages.

4.2 An Intuitive Approach to Stone Spaces

While readers familiar with Stone duality might want to skip this short interlude, we hope that the intuitive point of view provides help in grasping the technicalities of the definitions in the following sections. For a more in-depth presentation, we refer to [Joh86] or for a presentation condensed to the aspects important to us to [Geh11].

For an introduction, consider the following simple hypothetical situation:

Suppose you were asked whether the powerset Boolean algebra of $\{1, 2, \ldots, 25\}$ contained the element $\{2, 23\}$. Without even thinking (much), your answer would be “yes” and you would very likely have given it without having considered all 2^{25} elements of the powerset of $\{1, 2, \ldots, 25\}$ beforehand.

Of course, this observation is rather trivial, but it is an application of Stone duality. It is simple to give the answer to the question above, because we know the atomic elements the Boolean algebra is made of, which in this example are the singletons, of which we only need to consider 25, instead of 2^{25} elements of the Boolean algebra.

This explains the core idea behind Stone duality pretty well: It reduces a Boolean algebra to its so-to-say building blocks and allows us to answer questions, such as whether some set belongs to the Boolean algebra, more easily.

In computer science and specifically complexity theory whether some language belongs to some Boolean algebra of languages is a question frequently asked, but those Boolean algebras are in general much more complicated than Boolean algebras which are finite powersets and it is unclear, if atomic elements even exists. The regular languages, for instance, are not even isomorphic to some powerset algebra. This can easily be seen, when recalling that each powerset of a finite set is finite, but of an infinite set is automatically uncountable. The regular languages, however, are a countable Boolean algebra.

Even worse, there also are Boolean algebras that are uncountable and not isomorphic to a powerset Boolean algebra – for instance the class of languages recognised by formulas in $\mathbf{FO}[\mathcal{N}]$. That it is uncountable follows from the fact that there are uncountably many numerical predicates. One may verify that it is not isomorphic to a powerset Boolean algebra on the facts that it contains every singleton language but does not contain the language Parity.

We will now try to explain how Stone duality handles arbitrary Boolean algebras and can be seen as a way of identifying and examining the atomic elements of said Boolean algebras, starting from finite (powerset) Boolean algebras going to arbitrary ones.

Discrete Stone Duality

Recall that we use the notation borrowed from lattice theory: $\vee$ for meet, $\wedge$ for join and $\neg$ for complement. The greatest element of the Boolean algebra is denoted by 1 and the least element by 0. The choice of notation is to avoid confusion that the presented concepts are only viable for Boolean algebras of sets. However, due to Stones representation theorem, which states that every Boolean algebra is isomorphic to one of clopen sets, it is sufficient to think of Boolean algebras of sets, replacing $\vee$ by $\cup$, $\wedge$ by $\cap$ and $\neg$ by ^{c}.

In the following, let $\mathcal{B}$ be a finite Boolean algebra.

4.2.1 Definition

An element $x \in \mathcal{B}$ is called an *atom*, if $x \neq 0$ and $x = y \vee z$ for $y, z \in \mathcal{B}$ implies $y = x$ or $z = x$.

For instance, if X is a finite set, then the atoms of $\mathcal{P}(X)$ are the singletons and the set of atoms is thus isomorphic to X.

Atoms exist in every finite Boolean algebra: In particular, an element x is an atom if and only if it is not the zero element and for each $y \in \mathcal{B}$, either $x \leq y$ or x and y are disjoint, that is $x \wedge y = 0$. This implies that any two atoms x, y are disjoint.

Observing that every finite Boolean algebra is generated by its atoms, we obtain the even more general statement, as illustrated on an example in Figure 4.3:

4.2.2 Proposition

Any finite Boolean algebra is isomorphic to the powerset of its atoms.

Thus and to avoid introducing additional notation, we assume without loss of generality that we are dealing with finite powerset Boolean algebras for the meantime. Any finite set X should be thought of as the set of atoms of a finite Boolean algebra that is isomorphic to $\mathcal{P}(X)$.

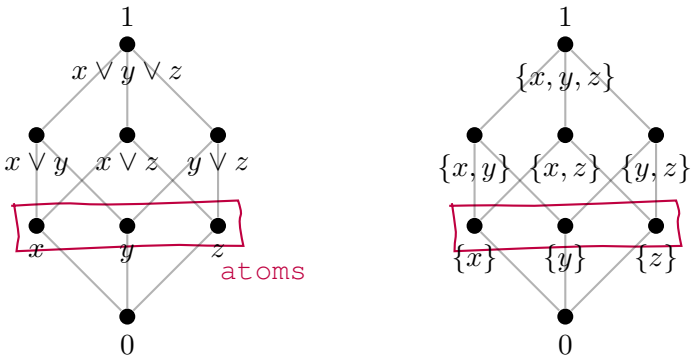


Figure 4.3: If x, y and z are disjoint, the Boolean algebra generated by them is isomorphic to the powerset of $\{x, y, z\}$.

Assume that X and Y are finite sets, then any map $f\colon X \to Y$ induces a morphism of Boolean algebras $f^{-1}\colon \mathcal{P}(Y) \to \mathcal{P}(X)$, given by $f^{-1}(S) = \{x \in X \mid f(x) \in S\}$ for $S \subseteq Y$.

Conversely given a morphism of Boolean algebras $f\colon \mathcal{P}(Y) \to \mathcal{P}(X)$, the function $f^{-1}\colon X \to Y$ sends $x \in X$ to the unique element $y \in Y$ such that $x \in f(\{y\})$. These two constructions are dual to each other in the sense that $(f^{-1})^{-1} = f$.

To understand them on a concrete example, one may verify that the map f below is a morphism of Boolean algebras and that f^{-1} is its dual map.

$$
\begin{aligned}
f\colon \mathcal{P}(\{1,2\}) &\to \mathcal{P}(\{1,2,3\}) \\
\emptyset &\mapsto \emptyset \\
\{1\} &\mapsto \{1\} \\
\{2\} &\mapsto \{2,3\} \\
\{1,2\} &\mapsto \{1,2,3\}
\end{aligned}
\qquad\qquad
\begin{aligned}
f^{-1}\colon \{1,2,3\} &\to \{1,2\} \\
1 &\mapsto \{1\} \\
2 &\mapsto \{2\} \\
3 &\mapsto \{2\}
\end{aligned}
$$

In particular, if $\iota\colon \mathcal{B} \hookrightarrow \mathcal{P}(X)$ is an inclusion, where $\mathcal{B}$ is a subalgebra of $\mathcal{P}(X)$, then the atoms of $\mathcal{B}$ induce a partition of X and ι^{-1} sends x to the unique partition element P, for which $x \in P$. Observe that, the canonical embedding $i\colon \mathcal{P}(\{1,2\}) \to \mathcal{P}(\{1,2,3\})$ is not a morphism of Boolean algebras and thus not an inclusion in the sense described before.

General Stone Duality

Observe that the previous findings, such as that every finite Boolean algebra is isomorphic to the powerset of its atoms, do not hold true on infinite Boolean algebras: The atoms of the regular languages are the singletons $\{w\}$ for $w \in A^*$. Hence **Reg** is not isomorphic to the powerset of its atoms, which is the set of all languages.

Since atoms are not sufficient, we will soon be confronted with a generalisation of these, which are called *ultrafilters*, that enable us again to jump back and forth between ultrafilters and a Boolean algebra, just like we could jump between elements of a finite set and their powerset. But first, we would like to motivate where the difference between finite and infinite case is roughly located and how we can draw a connection from the finite case to solve our problem with the infinite one.

Observe that the atoms for finite Boolean algebras are the smallest non-trivial elements with respect to inclusion. In the infinite case, these elements do not exist in general, as the following example illustrates:

4.2.3 Example

Let $\mathbb{N}_{\text{Co-Fin}}$ be the Boolean algebra over $\mathbb{N}$ that consists of all sets that are either finite or have a finite complement, that is

$$\mathbb{N}_{\text{Co-Fin}} = \{X \subseteq \mathbb{N} \mid X \text{ is finite or } \mathbb{N}\backslash X \text{ is finite.}\}.$$

The sets

$$X_{\geq n} = \{n, n+1, n+2, \dots\}$$

form a descending chain of sets $X_{\geq 1} \supseteq X_{\geq 2} \supseteq \dots$ in $\mathbb{N}_{\text{Co-Fin}}$, where all inclusions are proper.

In the finite case, proper descending chains always lead to the atoms below the element (see for instance Figure 4.3). Contrary to finite Boolean algebras, in infinite Boolean algebras there often exist elements that are *not* a finite Boolean combination of atoms, but have a proper descending chain of lesser elements not leading to an atom. The reason why we do not use descending chains to replace atoms is that they are not unique - even in the finite case, two different chains end up at the same atom.

The equivalent to atoms in the infinite case are *ultrafilters*. The core idea of ultrafilters is to generalise the term atom by something that is figuratively speaking alike to a search for an atom, which may never terminate, since we keep descending without ever finding an atom. Some searches however do reach an atom. Figure 4.4 may be

taken into account as an illustration of ultrafilters in the finite case, where each one hits an atom.

4.2.4 Definition

Let $\mathcal{B}$ be a Boolean algebra. An ultrafilter μ on $\mathcal{B}$ is a collection of elements of $\mathcal{B}$ such that

1. $0 \notin \mu$
2. $x \in \mu$ and $y \geq x$, then $y \in \mu$
3. $x, y \in \mu$, then $x \wedge y \in \mu$
4. for each $x \in \mathcal{B}$, either $x \in \mu$ or $\neg x \in \mu$

To make the conditions more plausible, we may replace them by some intuitive explanations from the analogy of a search for an atom: Condition 1. states that our search may never reach the bottom element – we either find an atom or we keep searching in lower elements. Condition 2. states, that if we suspect the atom to be below the element x, then it is also below any element larger than x. Condition 3. ensures that if we suspect the atom to be below x and below y, then it is also below $x \wedge y$ and finally condition 4. states that for any element $x \in \mathcal{B}$, the atom we are searching for has to be either below x or below its complement.

4.2.5 Proposition

In a finite Boolean algebra, the set of atoms is isomorphic (as a set: bijective) to the set of ultrafilters.

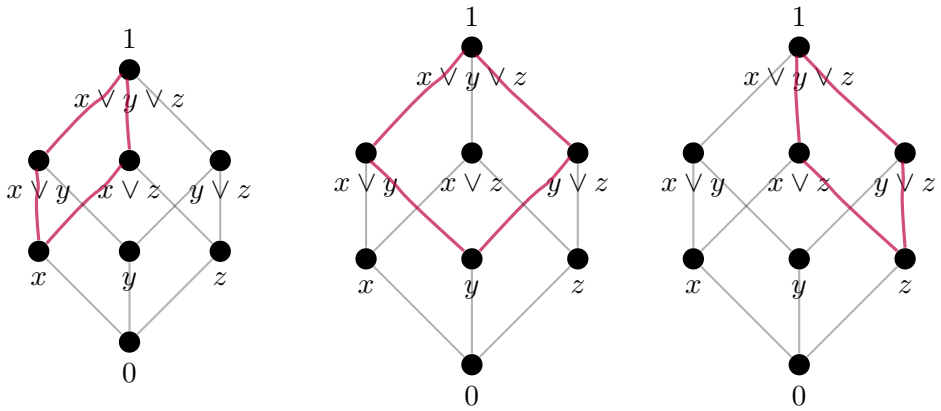


Figure 4.4: Ultrafilters searching for x, y and z respectively (left to right).

In Example 4.2.3, the ultrafilter on $\mathbb{N}_{\text{Co-Fin}}$ containing the chain $X_{\geq 1} \supseteq X_{\geq 2} \supseteq \ldots$ is the co-finite ultrafilter

$$\nu_\infty = \{X \in \mathbb{N}_{\text{Co-Fin}} \mid \mathbb{N} \backslash X \text{ is finite.}\}.$$

In fact, this ultrafilter is the only one that does not lead to an atom: all other ultrafilters must contain at least one set that is finite and hence are equal to

$$\nu_n := \{X \in \mathbb{N}_{\text{Co-Fin}} \mid \{n\} \in X\}$$

for some $n \in \mathbb{N}$.

We conclude, that the set of all ultrafilters of $\mathbb{N}_{\text{Co-Fin}}$ is the set $\mathbb{N} \cup \{\infty\}$, where ∞ represents the co-finite ultrafilter and the map sending any set $X \in \mathbb{N}_{\text{Co-Fin}}$ to

$$\widehat{X} := \{\nu \in \mathbb{N} \cup \{\infty\} \mid X \in \nu\}$$

is a morphism of Boolean algebras: For instance, let $X, Y \in \mathbb{N}_{\text{Co-Fin}}$, then by condition 2. the inclusion

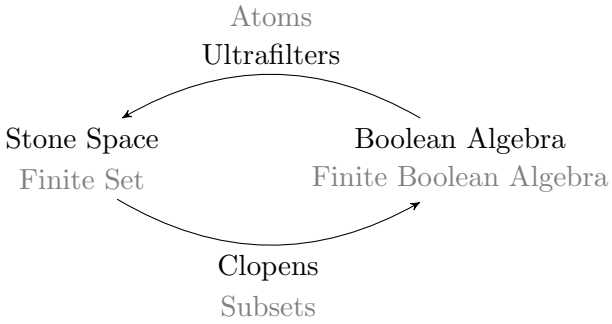
$$\underbrace{\{\nu \in \mathbb{N} \cup \{\infty\} \mid X \cap Y \in \nu\}}_{=\widehat{X \cap Y}} \subseteq \underbrace{\{\nu \in \mathbb{N} \cup \{\infty\} \mid X \in \nu \text{ and } Y \in \nu\}}_{=\widehat{X} \cap \widehat{Y}}$$

holds and similarly by condition 3. $\widehat{X} \cap \widehat{Y} \subseteq \widehat{X \cap Y}$. The cases for union and complementation follow in quite a similar manner. Hence, the structure of the Boolean algebra $\mathbb{N}_{\text{Co-Fin}}$ is preserved on the set of ultrafilters via the map associating $\widehat{X}$ to $X \in \mathbb{N}_{\text{Co-Fin}}$. We will see that the map sending X to $\widehat{X}$ is the foundation to generalising the correspondence between atoms and powerset Boolean algebras in the finite case by topological means.

To justify that already the finite case has connections to topology: The powerset of a set of atoms X is precisely the set of clopens on X, when X is equipped with the discrete topology. In that sense, we may equip the set of ultrafilters with a topology and obtain that the Boolean algebra is isomorphic to the set of clopens of said topology.

4.3 Stone Spaces

If we replace the terms from finite sets and finite algebras with those for infinite ones, we obtain the following picture, whose missing definitions, such as Stone space, we are going to fill in now.



4.3.1 Definition Stone Space

The Stone space of a Boolean algebra $\mathcal{B}$, denoted by $X_{\mathcal{B}}$ is the set of all ultrafilters over $\mathcal{B}$ together with the topology that is generated by the sets

$$\widehat{x} := \{\mu \in X_{\mathcal{B}} \mid x \in \mu\}$$

where $x \in \mathcal{B}$.

Remark: Often, the Stone space of a Boolean algebra is also referred to as the *dual space* of the Boolean algebra. For readers familiar with dual spaces commonly used in linear algebra, where the points are linear maps, this name might come as a slight surprise. However, there are some similarities, if we consider that ultrafilters over $\mathcal{B}$ and morphisms from $\mathcal{B}$ into the two element Boolean algebra $\{0,1\}$, are in one-to-one correspondence in the sense that for each ultrafilter μ, the map associating to each ultrafilter μ a morphism $v_\mu \colon \mathcal{B} \to \{0,1\}$ sending L to 1 if and only if $L \in \mu$ and the map associating to each morphism $v \colon \mathcal{B} \to \{0,1\}$ the ultrafilter $v^{-1}(\{1\})$, are inverse to each other.

Observe that for each $x \in \mathcal{B}$, the set $\widehat{x}$ is clopen, since

$$(\widehat{x})^c = \{\mu \in X_{\mathcal{B}} \mid x \notin \mu\} = \{\mu \in X_{\mathcal{B}} \mid \neg x \in \mu\} = \widehat{\neg x}.$$

In particular, denote by $\mathrm{Clopen}(X_{\mathcal{B}})$ the set of clopens of $X_{\mathcal{B}}$, then the map

$$\begin{aligned} \mathcal{B} &\to \mathrm{Clopen}(X_{\mathcal{B}}) \\ x &\mapsto \widehat{x} \end{aligned}$$

is a morphism of Boolean algebras, which follows, as indicated before, almost immediately from the axioms an ultrafilter satisfies. This isomorphism is precisely the one used to prove Stones representation theorem:

4.3.2 Theorem Stone (1936)

Every Boolean algebra is isomorphic to a Boolean algebra of sets.

In particular, each Stone space is compact, hausdorff and totally disconnected and each space having these properties is (isomorphic to) a Stone space.

Dual maps

In particular, Stone spaces preserve the properties that hold in the finite case for morphisms of Boolean algebras and their duals:

Let $\mathcal{B}$ and $\mathcal{C}$ be Boolean algebras and $f \colon \mathcal{B} \to \mathcal{C}$ a morphism of Boolean algebras, then f induces a continuous function $f^{-1} \colon X_{\mathcal{C}} \to X_{\mathcal{B}}$ in the following way: Let $x \in \mathcal{B}$ and $\nu \in X_{\mathcal{C}}$, then

$$x \in f^{-1}(\nu) \Leftrightarrow f(x) \in \nu.$$

Conversely, if $g\colon X_{\mathcal{C}} \to X_{\mathcal{B}}$ is a continuous function, it induces a morphism of Boolean algebras

$$\begin{aligned} g^{-1}\colon \operatorname{Clopen}(X_{\mathcal{B}}) &\to \operatorname{Clopen}(X_{\mathcal{C}}) \\ C &\mapsto \{\mu \in X_{\mathcal{C}} \mid g(\mu) \in C\}. \end{aligned}$$

and since the clopens of $X_{\mathcal{B}}$ (resp. $X_{\mathcal{C}}$) are isomorphic to $\mathcal{B}$ (resp. $\mathcal{C}$), g^{-1} is a morphism from $\mathcal{B}$ to $\mathcal{C}$. Again, those constructions are dual to each other in the sense that $(g^{-1})^{-1} = g$ and $(f^{-1})^{-1} = f$.

If f is surjective (resp. injective), then f^{-1} is injective (resp. surjective). This implies in particular, that if $\mathcal{B}$ is a subalgebra of $\mathcal{C}$, then $X_{\mathcal{B}}$ is a quotient of $X_{\mathcal{C}}$ (in the sense that it is a continuous image) and likewise if $\mathcal{B}$ is a quotient of $\mathcal{C}$, then $X_{\mathcal{B}}$ is a subspace of $X_{\mathcal{C}}$.

Hence, Stone spaces fully generalise discrete duality.

Stone Spaces and Algebra

As already mentioned, the Stone space of the regular languages is the free profinite monoid and we may reflect algebraic properties, such as aperiodicity via equations on the free profinite monoid, as Reiterman was able to prove.

But Stone spaces can also be found in even simpler contexts:

For instance, let L be a regular language and let $\mathcal{B}_L$ be the Boolean algebra generated by the sets $x^{-1}Ly^{-1}$, where $x, y \in A^*$. Then the elements of the Stone space $X_{\mathcal{B}_L}$ form a partition of A^* which is precisely the partition induced by the syntactic congruence of L. Thus, since L is regular, $X_{\mathcal{B}_L}$ is finite and equipped with the usual multiplication on the congruence classes the syntactic monoid of L.

Ultrafilter Equations

In the topological treatment of algebraic properties after Reiterman, equations of elements of the free profinite monoid play a key role. We have up to now eluded equations in the context of Stone duality, but in fact, Reiterman's observations fit perfectly in the framework. The precise connection is laid out in [GGP08]. Loosely stated on the example of the star-free languages it boils down to: The free profinite monoid is the Stone space of the regular languages and the star-free languages are a subalgebra of the regular languages. Hence, the Stone space of the star-free languages is a quotient of the free profinite monoid and the equation $x^\omega = x^{\omega+1}$ describes precisely the kernel of said quotient map.

This procedure of describing a Boolean algebra through the kernel of a quotient map between Stone spaces is possible in a much more general fashion for arbitrary non-regular Boolean algebras. Therefore, one particular Stone space is of special importance: the Stone space of the powerset Boolean algebra $\mathcal{P}(X)$ for a set X. We call that space the Stone-Čech-compactification of X and denote it by $\beta(X)$.

In particular, X embeds densely in $\beta(X)$ via the map $\iota_X \colon X \to \beta(X)$ sending x to the ultrafilter $\{y \in \mathcal{B} \mid x \leq y\}$. If $\mathcal{B}$ is a Boolean algebra of sets, this is equivalent to sending x to the ultrafilter $\{L \in \mathcal{B} \mid x \in L\}$.

This space is of such particular importance, since every Boolean algebra $\mathcal{B}$ is the subalgebra of some powerset Boolean algebra $\mathcal{P}(X)$ and hence there always exists a surjective continuous function $\pi_{\mathcal{B}} \colon \beta(X) \to X_{\mathcal{B}}$. Describing the kernel of said map $\pi_{\mathcal{B}}$ describes the Stone space of $\mathcal{B}$.

The elements of the kernel of $\pi_{\mathcal{B}}$ are what is commonly called the *equations* (or more precisely ultrafilter equations) for $\mathcal{B}$. That is for $\mu, \nu \in \beta(X)$, we write $[\mu \leftrightarrow \nu]$ and say that this is an equation for $\mathcal{B}$, if $\pi_{\mathcal{B}}(\mu) = \pi_{\mathcal{B}}(\nu)$.

According to previous observations, for each $x \in \mathcal{B}$ the equivalence $x \in \pi_{\mathcal{B}}(\mu)$ if and only if $\pi_{\mathcal{B}}^{-1}(x) \in \mu$ holds and since $\pi_{\mathcal{B}}$ is the dual of the inclusion, which consequently makes $\pi_{\mathcal{B}}^{-1}$ the inclusion, we obtain

$$\pi_{\mathcal{B}}(\mu) = \{x \in \mathcal{B} \mid x \in \mu\}.$$

Thus $[\mu \leftrightarrow \nu]$ is an equation for $\mathcal{B}$ if and only if for all $x \in \mathcal{B}$

$$x \in \mu \Leftrightarrow x \in \nu.$$

Additionally, X embeds densely in $X_{\mathcal{B}}$ through the map $\pi_{\mathcal{B}} \circ \iota_X$.

For instance, if L is a regular language and as before $\mathcal{B}_L$ is the Boolean algebra generated by all two-sided quotients of L, then $X_{\mathcal{B}_L}$ was the syntactic monoid of L and the embedding $\pi_{\mathcal{B}_L} \circ \iota_{A^*} \colon A^* \to X_{\mathcal{B}_L}$ is precisely the syntactic morphism of L.

Apart from providing us with equations, the Stone-Čech-compactification enjoys some other convenient properties, such as:

If X and Y are sets, then every map $f \colon X \to Y$ induces a continuous map $\beta f \colon \beta(X) \to \beta(Y)$ given by the equivalence

$$L \in \beta f(\mu) \Leftrightarrow f^{-1}(L) \in \mu$$

for each $L \subseteq Y$ and $\mu \in \beta(X)$. Moreover, the following useful proposition holds:

4.3.3 Proposition

If $f \colon X \to K$ is a function from a set X into a compact space K, then there exists a unique continuous extension $\widehat{f} \colon \beta(X) \to K$.

4.4 Conclusion

As we have seen, algebra and topology are closely related: On the one hand, the free profinite monoid, which is widely known as the completion of a metric space also has tight connections to algebra, be it through the metric that is itself defined via finite monoids or as the projective limit of finite monoids. On the other hand, it is also a

Stone dual space: namely that of the regular languages and hence an object that – in its primal definition – is of mainly topological nature. Similarly, each syntactic monoid of a regular language also is a Stone space.

Contrary to where the algebraic approach has its origins, in the regular languages, Stone duality is applicable also outside the regular world to arbitrary non-regular Boolean algebras.

However, since the algebraic approach was so fruitful, but yet scarcely applicable outside the regular languages, it seems only natural to enrich it with the topological perspective and see whether the previous findings on the regular languages, that relate algebra and topology, could add up to paint a larger picture outside the regular world.

Further Chapters

Hence the aim of the following chapters is to pursue that goal: On approach targets the visibly pushdown languages, a class slightly larger than the regular languages, which is still relatively good-natured in the sense that it forms a Boolean algebra and the underlying machine model has a decidable equivalence. For those, we diverge from monoids to finite structures with more operations than multiplication and then follow the same approach as in the regular languages, by constructing the completion of a metric space, obtaining the Stone space of the visibly pushdown languages. The second approach is applicable for arbitrary Boolean algebras of languages, but the main focus in this document lies on classes corresponding to fragments of logic. Here, metric spaces are in general not powerful enough, hence we use the more general concept of projective limits to generate topological objects relying on Stone duality. We then proceed to examine the block product, which was previously mainly employed on finite monoids and define it as a generalisation on those topological objects.

5

Visibly Pushdown Languages

Visibly Pushdown Languages were originally introduced by Alur and Madhusudan [AM04] as a subclass of the context free languages that enjoys similar good-natured closure properties, such as the regular languages. In the following, we are going to review the basic definitions and characterise the Stone space of the visibly pushdown languages in a similar way as the free profinite monoid is characterised – through finite algebraic objects.

5.1 Visibly Pushdown Automata

5.1.1 Definition Visibly Pushdown Alphabet

A *visible pushdown alphabet* is a finite alphabet A, which is partitioned into three sets A_C, A_R and A_I. Letters in A_C are often addressed as *call letters*, while letters in A_R are called *return letters* and letters in A_I *internal letters*.

In the following, let A be a visibly pushdown alphabet.

5.1.2 Definition Visibly pushdown automaton (VPA)

A *visibly pushdown automaton* is a tuple $(A, Q, q_0, \Gamma, \#, \delta, F)$, where

- A is a visibly pushdown alphabet,
- Q is a finite set, the set of *states*,
- $q_0 \in Q$ is the *initial state*,
- Γ is a finite alphabet, the *stack alphabet*,
- $\# \in \Gamma$ is the *bottom-of-stack symbol*,
- $\delta\colon A \times Q \times \Gamma \to Q \times \Gamma^*$ is the transition function with the following restrictions: For $q \in Q, a \in A$ and

  $$\delta(a, q, G) = (q', G'),$$

  where $G \in \Gamma$ and $G' \in \Gamma^*$, it must hold that
  - if $a \in A_C$, then $G' = G_0 G$, for some $G_0 \in \Gamma \backslash \{\#\}$.
  - if $a \in A_R$, then $G' = \lambda$.
  - if $a \in A_I$, then $G' = G$.
- and $F \subseteq Q$ is the set of final states.

Intuitively, the restrictions on the transition function can be interpreted as the letters of the visibly pushdown alphabet controlling the stack behavior. If a visibly pushdown automaton reads a call letter, it pushes to the stack, if it reads a return letter, it pops from the stack and reading an internal letter leaves the stack untouched.

5.1.3 Definition Language of a VPA

Let $M = (A, Q, q_0, \Gamma, \#, \delta, F)$ be a visibly pushdown automaton and let $k \in \mathbb{N}$ and $G_i \in \Gamma$ for $i = 1, \ldots, k$. We define the *extended transition function*, denoted by $\widehat{\delta}\colon A^* \times Q \times \Gamma^* \to Q \times \Gamma^*$, inductively as

- $\widehat{\delta}(\lambda, q, G_0 \ldots G_k) = (q, G_0 \ldots G_k)$
- $\widehat{\delta}(aw, q, G_0 \ldots G_k) = \widehat{\delta}(w, q', G' G_1 \ldots G_k)$, where $(q', G') = \delta(a, q, G_0)$.

The language accepted by M is the language

$$L(M) = \{w \in A^* \mid \widehat{\delta}(w, q_0, \#) \in F \times \{\#\}\}$$

Consequently, a language $L \subseteq A^*$ is a *visibly pushdown language (VPL)* if it is accepted by a visibly pushdown automaton.

Let a be a call, b a return and c a neutral letter. Some instances of visibly pushdown languages are: The language $\{a^n b^n \mid n \in \mathbb{N}\}$, the Dyck language over any finite set

of parentheses or the language accepted by the grammar with the production rules

$$S \to acSb \mid aScb \mid \lambda,$$

which we will in the following call the Ludwig language. [Lud18]

Observe that the alphabet controlling the stack behaviour of the automaton and the condition that the automaton accepts only at stack-height zero, results in the property that each word in a VPL has a matching between call and return letters, as illustrated in Figure 5.1. For each call letter pushing a symbol to the stack, there is a matching return letter removing it from the stack. This is a slight difference from the original definition of VPA in [AM04].

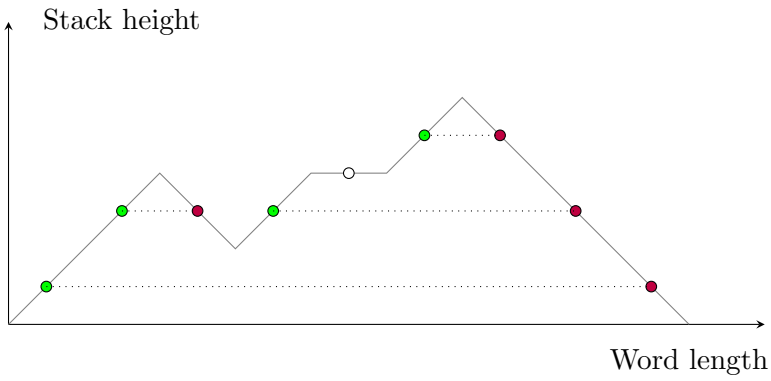


Figure 5.1: Height profile of a well matched word. Call letters indicated in green, return letters in red and internal letters in white. The dotted lines indicate the matching.

Independent of any VPA, we define the set of so-called *well-matched* words over A, denoted by $A^{\triangle}$ inductively:

- The empty word λ is well-matched
- Each $c \in A_I$ is well-matched.
- For a well-matched word w, $a \in A_C$ and $b \in A_R$, the word awb is well-matched.
- If u, v are well-matched, then so is uv.

For instance, let $A_C = \{a, c\}$, $A_R = \{b, d\}$ and $A_I = \{e\}$, then the word $aacbbed$ is well-matched. Note that the return letter matching the call letter a need not always be the same return letter, but may be any one of the return letters.

If $M = (A, Q, q_0, \Gamma, \#, \delta, F)$ is a VPA, we denote by $\pi_Q \colon Q \times \Gamma^* \to Q$ the projection onto the state. As a generalisation of the previous observation on the stack-behaviour of well-matched words, we obtain:

5.1.4 Lemma

Let $w \in A^{\triangle}$ and let $M = (A, Q, q_0, \Gamma, \#, \delta, F)$ be a VPA. Then

$$\widehat{\delta}(w, q, G) = \pi_Q(\widehat{\delta}(w, q, G)) \times \{G\}$$

The proof is straight-forward.

5.2 VPL in Terms of Algebra

VPL were already characterised in [AKMV05] through finite congruences. We adapt that result slightly to work with well-matched VPL and enrich it with additional algebraic structure: For instance, the set of all well-matched words $A^{\triangle}$ is a submonoid of A^* that additionally has an operation sending the word w to awb for a a call and b a return letter. One may visualise this operation, as an extension in height of the chain of mountains that is the height profile of a well-matched word.

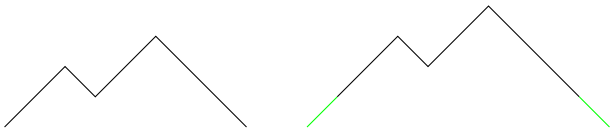


Figure 5.2: Well-matched word w on the left and awb on the right.

Generalising from that specific instance, we obtain:

5.2.1 Definition

An Ext-algebra is a set R together with a multiplication $\cdot$ and a set $\mathcal{O}(R)$ of unary operations from R to R, such that $(R, \cdot)$ and $(\mathcal{O}(R), \circ)$ are monoids ($\circ$ is the composition) and the maps $x \mapsto r \cdot x$ and $x \mapsto x \cdot r$ are contained in $\mathcal{O}(R)$ for each $r \in R$. We usually omit to mention $\mathcal{O}(R)$ and say that R is an Ext-algebra.

Note on forest algebras. Each Ext-algebra $(R, \mathcal{O}(R))$ is a forest algebra as introduced in [BW08], where the horizontal monoid is R, the vertical monoid $\mathcal{O}(R)$ and the action of $\mathcal{O}(R)$ on R is function application. We distinguish them, since we are investigating languages of words rather than languages of trees. Still, it should be mentioned that VPL and regular tree languages have very close connections [AM04].

Observe that the set of all well-matched words $A^{\triangle}$ is an Ext-algebra: For any two words $u, v \in A^*$ such that $uv \in A^{\triangle}$ and $x \in A^{\triangle}$, let $\mathrm{ext}_{u,v}(x) = uxv$. Then we let $\mathcal{O}(A^{\triangle})$ be the set of all maps $\mathrm{ext}_{u,v}$. The left- and right multiplication maps ($x \mapsto x \cdot r$ and $x \mapsto r \cdot x$) are given by $\mathrm{ext}_{w,\lambda}$ (resp. $\mathrm{ext}_{\lambda,w}$) for $w \in A^{\triangle}$.

5.2.2 Definition Morphism

Let R and S be EXT-algebras. A morphism from R to S is a tuple of monoid morphisms (g, h) with $g\colon R \to S$ and $h\colon \mathcal{O}(R) \to \mathcal{O}(S)$ such that for all $e \in \mathcal{O}(R)$ and $r \in R$: $h(e)(g(r)) = g(e(r))$.

Observe that g is implicitly determined by h, since g is monoid morphism and hence $g(1_R) = 1_S$. For $r, x \in R$ letting $m_r(x) = r \cdot x$, we obtain

$$g(r) = g(m_r(1_R)) = h(m_r)(1_S).$$

Hence we cease to distinguish between g and h and say that $h\colon R \to S$ is a morphism of EXT-algebras.

In particular, by the inductive definition of the well-matched words, any morphism $h\colon A^{\triangle} \to R$ is uniquely determined by its values on $\mathrm{ext}_{a,b}$ for $a \in A_C$ and $b \in A_R$, $\mathrm{ext}_{c,\lambda}$ and $\mathrm{ext}_{\lambda,c}$ for $c \in A_I$.

5.2.3 Definition

Let R, S be EXT-algebras. A morphism (g, h) from R to S is called surjective (resp. injective) if both $g\colon R \to S$ and $h\colon \mathcal{O}(R) \to \mathcal{O}(S)$ are surjective (resp. injective).
We say that S is a sub of R, if $S \subseteq R$ and $\mathcal{O}(S) \subseteq \mathcal{O}(R)$. Similarly S is a quotient of R, if there exists a surjective morphism from R to S and finally, S divides R if S is the quotient of a sub of R.

In the following, we often write $\mathrm{ext}_{a,b}$ for the operation on $A^{\triangle}$ and also $\mathrm{ext}_{a,b}$ for an operation on some EXT-algebra R. This often has its origin in the fact that the $\mathrm{ext}_{a,b}$ operation on R is considered the morphic image of $\mathrm{ext}_{a,b}$ on $A^{\triangle}$ for some particular morphism $h\colon A^{\triangle} \to R$. We assume that it is understood from the context, which is which.

By that convention, a morphism $h\colon A^{\triangle} \to R$ into an EXT-algebra satisfies

$$h(awb) = \mathrm{ext}_{a,b}(h(w)).$$

Apart from the well-matched words $A^{\triangle}$, all other EXT-algebras we will be regarding are finite. Multiplication and unary operations are represented as tables, as displayed on the following example.

5.2.4 Example

The following tables display an Ext-algebra, where A is divided into $A_C = \{a\}$ and $A_R = \{b\}$ and A_I is empty. The multiplication is displayed on the left and $\text{ext}_{a,b}$ operation on the right.

$\cdot$	1	x	0
1	1	x	0
x	x	0	0
0	0	0	0

	1	x	0
$\text{ext}_{a,b}$	x	x	0
$\text{ext}_{ab,\lambda}$	x	0	0
$\text{ext}_{ab,ab}$	0	0	0

Observe that the names of the operations may be replaced by other representatives. For instance $\text{ext}_{a,b} = \text{ext}_{aa,bb}$ and $\text{ext}_{ab,\lambda} = \text{ext}_{aabb,\lambda}$.

Language Reconition

Similar to recognition of regular languages by monoid morphisms, we can recognise languages of well-matched words via Ext-algebra morphisms. While the syntactic monoid of a VPL, such as $\{a^n b^n \mid n \in \mathbb{N}\}$ is in general infinite, our notion of recognition through algebras instead of monoids and in particular the additional algebraic structure of Ext-algebras allows us to obtain finite recognising objects for the non-regular VCL. This leads to the main theorem at the end of the section, stating that VPL are precisely the languages recognised by finite Ext-algebras.

5.2.5 Definition Language recognition

A language $L \subseteq A^{\triangle}$ is recognised by an Ext-algebra R, if there is a morphism $h\colon A^{\triangle} \to R$, such that $L = h^{-1}(T)$, for some $T \subseteq R$. Equivalently $L = h^{-1}(h(L))$.

For instance, let a be a call and b a return letter. Then the Ext-algebra from Example 5.2.4, recognises the language $\{a^n b^n \mid n \in \mathbb{N}\}$.

5.2.6 Example

The Ludwig language is recognised by the Ext-algebra

$\cdot$	1	c	acb	$acbc$	0
1	1	c	acb	$acbc$	0
c	c	0	$acbc$	0	0
acb	acb	$acbc$	0	0	0
$acbc$	$acbc$	0	0	0	0
0	0	0	0	0	0

	1	c	acb	$acbc$	0
$\text{ext}_{a,b}$	0	acb	0	acb	0
$\text{ext}_{\lambda,c}$	c	0	acbc	0	0
$\text{ext}_{a,cb}$	acb	0	acb	0	0
$\text{ext}_{ac,cb}$	0	0	0	0	0

via the morphism h – which is given by the names of the operations. Then $L = h^{-1}(\{acb\})$. Observe that $\text{ext}_{c,\lambda} = \text{ext}_{\lambda,c}$.

5.2.7 Example

As a second example, consider the language H^+ [Hah15] over the alphabet $A = \{a, b\}$, where a is a call and b a return letter. This language is given by the production rules

$$S \to aNb \mid SS \mid \lambda,\ N \to aSb \mid NN \mid NS \mid SN.$$

Intuitively speaking, the language encodes all true Boolean formulae in the sense that the empty word is considered true, concatenation is conjunction and enclosing a word by a and b is negation. Then H^+ is recognised by the Ext-algebra R_{H^+} defined below.

$\cdot$	1	0
1	1	0
0	0	0

	1	0
$\mathrm{ext}_{a,b}$	0	1
ext_{a^2,b^2}	1	0
$\mathrm{ext}_{ab,\lambda}$	0	0
$\mathrm{ext}_{a^2b,b}$	1	1

As an intermediate step towards the main theorem, in which we prove that languages recognised by Ext-algebras are precisely the VPL, we show that for each language of well-matched words, there exists a minimal Ext-algebra recognising it.

We say that an equivalence relation $\sim$ on an Ext-algebra R is a congruence of Ext-algebras if and only if for all $e \in \mathcal{O}(R)$:

$$x \sim y \Leftrightarrow e(x) \sim e(y).$$

In the following, by congruence, we mean congruence of Ext-algebras unless explicitly stated otherwise.

Observe that the map $x \mapsto [x]$, where $[x]$ denotes the equivalence class of x with respect to $\sim$ is a morphism of Ext-algebras, in the sense that its image is equipped with the operations $e([x]) = [e(x)]$ for each $e \in \mathcal{O}(R)$. We denote that Ext-algebra by $R/_\sim$.

5.2.8 Definition Syntactic congruence

Let $L \subseteq A^\triangle$. We say that two words $u, v \in A^\triangle$ are equivalent with respect to L and write $u \sim_L v$ if for all $x, y \in A^\triangle$, the equivalence

$$xuy \in L \Leftrightarrow xvy \in L$$

and for all $a \in A_C$ and $b \in A_R$

$$\mathrm{ext}_{a,b}(u) \in L \Leftrightarrow \mathrm{ext}_{a,b}(v) \in L$$

holds.

Observe that for well-matched words u and v, this implies that if $x \sim_L y$ then the equivalence $\mathrm{ext}_{u,v}(x) = uxv \in L \Leftrightarrow \mathrm{ext}_{u,v}(y) = uyv \in L$ holds, but the converse does not hold in general.

The relation $\sim_L$ defines a congruence on $A^\triangle$ and therefore $A^\triangle/\sim_L$ is an Ext-algebra with the canonical operations on the equivalence-classes. We call $\sim_L$ the syntactic congruence of L and $A^\triangle/\sim_L$ the syntactic Ext-algebra of L, denoted by $\mathrm{Ext}(L)$.

5.2.9 Example

Let a be a call and b a return letter. Then, the syntactic Ext-algebra of $(ab)^*$ is given by

$\cdot$	1	ab	0
1	1	ab	0
ab	ab	ab	0
0	0	0	0

	1	ab	0
$\mathrm{ext}_{a,b}$	ab	0	0
$\mathrm{ext}_{ab,\lambda}$	ab	ab	0
$\mathrm{ext}_{\lambda,ab}$	ab	ab	0

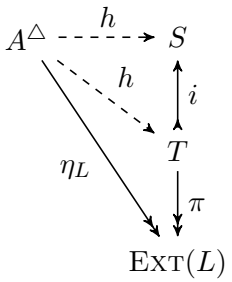
One may prove that division defines a partial order on finite Ext-algebras. The proof of this statement is almost entirely the same as for finite monoids and thus omitted. Also similar to finite monoids, we show that there exists a smallest recognising object.

5.2.10 Proposition

A language $L \subseteq A^\triangle$ is recognised by an Ext-algebra S, if and only if $\mathrm{Ext}(L)$ divides S.

Proof. Observe that for both directions of the proof, we may assume that S is an Ext-algebra.

Let us first prove the converse direction and suppose that S is some Ext-algebra that is divided by $\mathrm{Ext}(L)$. Hence there exists a subalgebra T of S and a surjective morphism $\pi\colon T \to \mathrm{Ext}(L)$. By η_L denote the syntactic morphism of L. We show the existence of a morphism $h\colon A^\triangle \to S$ such that the following diagram commutes.



Define the morphism $h\colon A^\triangle \to T$, by choosing $h(\mathrm{ext}_{\lambda,c}) \in \pi^{-1}(\eta_L(\mathrm{ext}_{\lambda,c}))$ and $h(\mathrm{ext}_{a,b}) \in \pi^{-1}(\eta_L(\mathrm{ext}_{a,b}))$ for all $c \in A_I$ and $a \in A_C, b \in A_R$. Since π is surjective,

such elements exists. Since T is a subalgebra of S, we can view h as a morphism from $A^{\triangle}$ to S. By the choice of h, we have for $w \in A^{\triangle}$

$$\pi(h(w)) = \prod_{i=1}^{|w|} \pi(h(w_i)) = \prod_{i=1}^{|w|} \eta(w_i) = \eta(w)$$

and hence, by $L = \eta^{-1}(\eta(L))$

$$w \in L \Leftrightarrow \eta(w) \in \eta(L) \Leftrightarrow \pi(h(w)) \in \eta(L) \Leftrightarrow h(w) \in \pi^{-1}(\eta(L)).$$

Thus $L = h^{-1}(\pi^{-1}(\eta(L)))$ and S recognises L.

Now assume that S is some EXT-algebra that recognises L by the morphism $h\colon A^{\triangle} \to S$. Then the image of h is a subalgebra of S. We show that $h(S)$ has $\text{EXT}(L)$ as a quotient. For that, we prove that η factors through h, that is for any two $u, v \in A^{\triangle}$, $h(u) = h(v)$ implies $\eta(u) = \eta(v)$. If $h(u) = h(v)$, since h is a morphism recognising L, we obtain $xuy \in L$ iff $xvy \in L$ and $\text{ext}_{a,b}(u) \in L$ iff $\text{ext}_{a,b}(v) \in L$ and hence $\eta(u) = \eta(v)$. Thus we can define $\pi\colon h(S) \to \text{EXT}(L)$ by letting $\pi(s) = \eta(h^{-1}(s))$. One can then verify that π is indeed a morphism and thus $\text{EXT}(L)$ is a quotient of S. ■

5.2.11 Definition

Let R and S be two EXT-algebras, then their direct product is the set $R \times S$ together with the operations for $(r, s) \in R \times S$

$$e_R(r, s) = (e_R(r), s) \text{ for } e_R \in \mathcal{O}(R)$$

and accordingly

$$e_S(r, s) = (r, e_S(s)) \text{ for } e_S \in \mathcal{O}(S)$$

Just as for the regular languages, where the product of finite monoids recognises Boolean combinations of the languages recognised by the components of the product, the same holds for EXT-algebras.

5.2.12 Proposition

Let R and S be two EXT-algebras then the product $R \times S$ recognises precisely the languages that are Boolean combinations of languages recognised by R or S.

The proof is entirely straight-forward and thus omitted.

The following proposition is a translation of the result by Alur et. al. [AKMV05] to EXT-algebras.

5.2.13 Proposition

A language $L \subseteq A^{\triangle}$ is VPL, if and only if it is recognised by a finite EXT-algebra.

Proof. Let $L \subseteq A^{\triangle}$ be a VPL and let $M_L = (A, Q, q_0, \Gamma, \#, \delta, F)$ be a VPA accepting L. Recall that by $\pi_Q \colon Q \times \Gamma^* \to Q$, we denote the projection to the state. We will now define an equivalence on well matched words, based on the states of the automaton M_L. Let $w \in A^{\triangle}, G \in \Gamma$ and define the function

$$\begin{aligned} f_{w,G} \colon Q &\to Q \\ q &\mapsto \pi_Q(\widehat{\delta}(w, q, G)). \end{aligned}$$

Observe that for $w_1, w_2 \in A^{\triangle}$ the relation

$$w_1 \sim_{M_L} w_2 \text{ iff for all } G \in \Gamma,\ f_{w_1,G} = f_{w_2,G}$$

is an equivalence relation on $A^{\triangle}$. Note that it is also a congruence on $A^{\triangle}$, since for $w_1, w_2, z \in A^{\triangle}$ with $w_1 \sim_{M_L} w_2$ and $G \in \Gamma$, we have

$$\begin{aligned} \pi_Q(\widehat{\delta}(zw_1, q, G)) &= \pi_Q(\widehat{\delta}(w_1, \widehat{\delta}(z, q, G))) \\ &= \pi_Q(\widehat{\delta}(w_1, \widehat{\delta}(\lambda, q', G))) \text{ for some } q', \text{ since } z \text{ is well-matched,} \\ &= \pi_Q(\widehat{\delta}(w_1, q', G)) \\ &= \pi_Q(\widehat{\delta}(w_2, q', G)) \text{ since } f_{w_1,G} = f_{w_2,G}, \\ &= \pi_Q(\widehat{\delta}(w_2, \widehat{\delta}(z, q, G))) \\ &= \pi_Q(\widehat{\delta}(zw_2, q, G)). \end{aligned}$$

The case for $w_1 z$ and $w_2 z$ follows from Lemma 5.1.4. Combining the two yields $xw_1y \sim_{M_L} xw_2y$ for $x, y \in A^{\triangle}$. Moreover

$$\begin{aligned} \pi_Q(\widehat{\delta}(aw_1b, q, G)) &= \pi_Q(\widehat{\delta}(w_1b, \delta(a, q, G))) \\ &= \pi_Q(\widehat{\delta}(w_1b, q', G_aG)) \text{ for some } q' \in Q, G_a \in \Gamma, \\ &= \pi_Q(\widehat{\delta}(b, \pi_Q(\widehat{\delta}(w_1, q', G_a)), G_aG)) \text{ by Lemma 5.1.4,} \\ &= \pi_Q(\widehat{\delta}(b, \pi_Q(\widehat{\delta}(w_2, q', G_a)), G_aG)) \text{ since } f_{w_1,G_a} = f_{w_2,G_a}, \\ &= \pi_Q(\widehat{\delta}(w_2b, q', G_aG)) \\ &= \pi_Q(\widehat{\delta}(w_2b, \delta(a, q, G))) \\ &= \pi_Q(\widehat{\delta}(\mathrm{ext}_{a,b}(w_2), q, G)) \end{aligned}$$

Since Γ is finite and there are $|Q|^{|Q|}$ different functions from Q to Q, $\sim_{M_L}$ has at most $|\Gamma| \cdot |Q|^{|Q|}$ congruence classes and $\sim_{M_L}$ is finite, which also makes $A^{\triangle}\backslash_{\sim_{M_L}}$ finite. By construction if $w_1 \sim_{M_L} w_2$, then $f_{w_1,\#}(q_0) = f_{w_2,\#}(q_0)$ and hence

$$w_1 \in L \Leftrightarrow \pi_Q(\widehat{\delta}(w_1, q_0, \#)) \in F \Leftrightarrow \pi_Q(\widehat{\delta}(w_2, q_0, \#)) \in F \Leftrightarrow w_2 \in L,$$

which implies that $\sim_{M_L}$ is a refinement of the syntactic congruence, which in turn implies that the syntactic Ext-algebra of L divides $A^{\triangle}/_{\sim_{M_L}}$. Thus L is recognised by a finite Ext-algebra by Proposition 5.2.10.

For the converse direction, assume that L is recognised by a finite Ext-algebra R via a morphism $h\colon A^{\triangle} \to R$. We construct a visibly pushdown automaton $M = (A, Q, q_0, \Gamma, \#, \delta, F)$ recognising L as follows

- $Q = R$
- $q_0 = h(\lambda)$
- $\Gamma = \{\#\} \cup (Q \times A_C)$
- $\delta\colon A \times Q \times \Gamma \to Q \times \Gamma^*$ is defined as follows: Let $a, a' \in A$ and $q, q' \in Q$. Then

$$\delta(a, q, (q', a')) = \begin{cases} (h(\lambda), (q, a)(q', a')) & \text{if } a \in A_C, \\ (q' \cdot \mathrm{ext}_{a'a}(q), \lambda) & \text{if } a \in A_R \text{ and} \\ (q \cdot h(a), (q', a')) & \text{if } a \in A_I \end{cases}$$

- $F = h(L)$.

We have to show that for each $w \in A^{\triangle}$, $\pi_Q(\widehat{\delta}(w, q_0, \#)) \in F$ if and only if $w \in L$. We prove this by showing that $\widehat{\delta}(w, q, G) = (q \cdot h(w), G)$ for each $G \in \Gamma^*$ by induction on the structure of words in $A^{\triangle}$.

Inductive start: Let $w \in A_I$ then

$$\widehat{\delta}(w, q, G) = \delta(w, q, G) = (q \cdot h(w), G).$$

Inductive step: Let $w, w_1, w_2 \in A^{\triangle}$ be some words for which the claim holds. Then

$$\begin{aligned} \widehat{\delta}(w_1 \cdot w_2, q, G) &= \widehat{\delta}(w_2, \widehat{\delta}(w_1, q, G) \\ &= \widehat{\delta}(w_2, q \cdot h(w_1), G) \\ &= (q \cdot h(w_1) \cdot h(w_2), G) \\ &= (q \cdot h(w_1 \cdot w_2), G) \end{aligned}$$

and

$$\begin{aligned} \widehat{\delta}(awb, q, G) &= \widehat{\delta}(wb, \delta(a, q, G)) \\ &= \widehat{\delta}(wb, h(\lambda), (q, a)G) \\ &= \widehat{\delta}(b, \pi_Q(\widehat{\delta}(w, h(\lambda), (q, a))), G) \\ &= \widehat{\delta}(b, h(w), (q, a)G) \\ &= \delta(b, h(w), (q, a)G) \\ &= (q \cdot \mathrm{ext}_{a,b}(h(w)), G) \\ &= (q \cdot h(awb), G) \end{aligned}$$

Setting $q = q_0 = h(\lambda)$ proves the claim. ∎

5.3 An Eilenberg Theorem

In this section, we show that there is a one-to-one correspondence between classes of VPL and classes of Ext-algebras with certain closure properties.

Note. To readers familiar with universal algebra, these closure properties should come as no surprise, see pseudo-varieties in [Alm95]. However, to keep the subject accessible to a broader community, we refrain from using the slang of universal algebra. It should however be mentioned, that it might be possible to obtain these results using category theoretic machinery as, for instance, in [UACM16] or [Boj15].

We define the following operations on well-matched words: If $L \subseteq A^{\triangle}$ is a language of well-matched words and $u, v \in A^*$ words such that $uv \in A^{\triangle}$, then

$$\mathrm{ext}_{u,v}^{-1}(L) = \{w \in A^{\triangle} \mid \mathrm{ext}_{u,v}(w) \in L\},$$

which we call an inverse Ext-operation. Observe that, for instance, $\mathrm{ext}_{w,\lambda}^{-1}(L)$ for $w \in A^{\triangle}$ is very similar to what is known under the name of quotients by words for languages over A^*.

5.3.1 Definition Pseudo-Variety of VPL

A pseudo-variety of visibly pushdown languages is a class $\mathcal{V}$ of languages of well-matched words such that

1. for each visibly pushdown alphabet A, the set $\mathcal{V}(A)$ is a Boolean algebra of VPL over $A^{\triangle}$,
2. the set $\mathcal{V}(A)$ is closed under inverse extend operations, that is for $L \in \mathcal{V}(A)$ and $u, v \in A^*$ such that $uv \in A^{\triangle}$, $\mathrm{ext}_{u,v}^{-1}(L)$ is an element of $\mathcal{V}(A)$,
3. and $\mathcal{V}$ is closed under inverse morphisms, that is if $h\colon A^{\triangle} \to B^{\triangle}$ is a morphism of Ext-algebras, then $L \in \mathcal{V}(B)$ implies $h^{-1}(L) \in \mathcal{V}(A)$.

For instance, the set of all languages of the form $L \cap A^{\triangle}$, where L is some regular language, forms a pseudo-variety (of VPL) – we drop the additional term, if it is understandable from the context. The visibly pushdown languages contained in $\mathbf{AC}^0$, however, do not form a pseudo-variety, since the Dyck language $\mathbb{D}$ over any set of parentheses and hence the set of all well-matched words $A^{\triangle}$ is not in $\mathbf{AC}^0$.

5.3.2 Proposition

Let $\mathcal{V}$ be a pseudo-variety of VPL, $L \in \mathcal{V}(A)$ and $\eta_L\colon A^{\triangle} \to R$ the syntactic morphism. Then for each $x \in R$, $\eta_L^{-1}(x) \in \mathcal{V}(A)$.

Proof. Let K be the syntactic image of L, that is $K = \eta_L(L)$. Then $L = \eta_L^{-1}(K)$. We prove that we can express x as a Boolean combination of quotients and inverse extend operations of K. Define the sets

$$C = \bigcap_{\substack{u,v \in A^* : uv \in A^\triangle \\ x \in \mathrm{ext}_{u,v}^{-1}(K)}} \mathrm{ext}_{u,v}^{-1}(K)$$

and

$$N = \bigcup_{\substack{u,v \in A^* : uv \in A^\triangle \\ x \notin \mathrm{ext}_{u,v}^{-1}(K)}} \mathrm{ext}_{u,v}^{-1}(K).$$

It is not hard to see, that $u \in C \backslash N$ if and only if $u \sim_L x$. Hence

$$C \backslash N = \{x\}.$$

Since $\eta_L^{-1}(K) \in \mathcal{V}(A)$ and the preimage of a morphism commutes both with quotients by words and inverse extend operations, we obtain $\eta_L^{-1}(\{x\}) \in \mathcal{V}(A^*)$. ∎

We now introduce the notion of pseudo-variety of Ext-algebras that is necessary to obtain an Eilenberg-like one-to-one correspondence between those and pseudo-varieties of VPL.

5.3.3 Definition

A class $\mathbf{V}$ of Ext-algebras is called a pseudo-variety, if it is closed under

1. division and
2. finite direct products.

Since the following proofs do not differ significantly from the ones that give the original Eilenberg theorem between pseudo-varieties of regular languages and pseudo-varieties of finite monoids, we keep them short.

5.3.4 Proposition

Let $\mathbf{V}$ be a pseudo-variety of Ext-algebras, then the languages recognised by $\mathbf{V}$ form a pseudo-variety of well-matched languages.

Proof. Denote by $\mathcal{V}(A)$ the set of all languages over $A^\triangle$ that are recognised by elements of $\mathbf{V}$. Since $\mathbf{V}$ is closed under direct products, it follows that $\mathcal{V}(A)$ is a Boolean algebra.

Let $R \in \mathbf{V}$ and $h \colon A^\triangle \to R$ a morphism with $L = h^{-1}(K)$ for some $K \subseteq R$. It is straight-forward that for any $w \in A^\triangle$

$$w^{-1}L = h^{-1}\left(\{m \in R \mid h(w)m \in K\}\right)$$

and hence $w^{-1}L \in \mathcal{V}(A)$. It follows from a similar argument, that also $Lw^{-1} \in \mathcal{V}(A)$ and $\mathrm{ext}_{a,b}^{-1}(L) \in \mathcal{V}(A)$.

Also, $\mathcal{V}$ is closed under inverse morphisms, for the reason that if $h\colon A^{\triangle} \to R$ recognises L, then $h \circ \varphi$ with $\varphi\colon A^{\triangle} \to B^{\triangle}$ recognises $\varphi^{-1}(L)$.

■

We denote the correspondence sending a pseudo-variety of Ext-algebras to a pseudo-variety of VPL by $\mathbf{V} \to \mathcal{V}$, where a pseudo-variety of Ext-algebras maps to the pseudo-variety of all languages recognised by members of $\mathbf{V}$. If some pseudo-variety $\mathbf{V}$ maps to $\mathcal{V}$ via the correspondence, we write $\mathbf{V} \mapsto \mathcal{V}$.

5.3.5 Proposition

The correspondence $\mathbf{V} \to \mathcal{V}$ is one-to-one.

Proof. Assume that $\mathbf{V}$ and $\mathbf{W}$ are two pseudo-varieties of Ext-algebras with $\mathbf{V} \mapsto \mathcal{V}$ and $\mathbf{W} \mapsto \mathcal{V}$. For an Ext-algebra $R \in \mathbf{V}$ and morphism $h\colon A^{\triangle} \to R$ and any $m \in R$, define the language $L_m = h^{-1}(m)$. Observe that by 5.2.10, the syntactic monoid $\mathrm{Ext}(L_m)$ is contained in $\mathbf{V}$ and $\mathbf{W}$. Also, R divides

$$\prod_{m \in R} \mathrm{Ext}(L_m),$$

which results in $R \in \mathbf{W}$. By the symmetry of the argument $\mathbf{V} = \mathbf{W}$. ■

To each pseudo-variety $\mathcal{V}$ of VPL, associate the pseudo-variety of Ext-algebras generated by all syntactic Ext-algebras of languages $L \in \mathcal{V}(A)$ for some visibly pushdown alphabet A. Denote that correspondence by $\mathcal{V} \to \mathbf{V}$. As before, $\mathcal{V} \mapsto \mathbf{V}$ indicates that $\mathcal{V}$ corresponds to $\mathbf{V}$ via $\mathcal{V} \to \mathbf{V}$.

5.3.6 Theorem

The correspondences $\mathcal{V} \to \mathbf{V}$ and $\mathbf{V} \to \mathcal{V}$ are mutually inverse bijections.

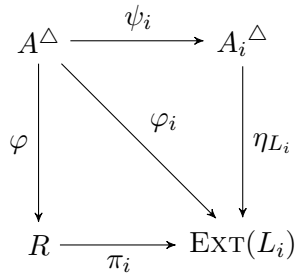
Proof. Let $\mathcal{V}$ be a pseudo-variety of VPL with $\mathcal{V} \mapsto \mathbf{V}$ and $\mathbf{V} \mapsto \mathcal{W}$. We show that $\mathcal{V} = \mathcal{W}$.

Let $L \in \mathcal{V}(A)$, then the syntactic Ext-algebra of L is contained in $\mathbf{V}$ and hence also $L \in \mathcal{W}(A)$.

For the converse direction assume that $L \in \mathcal{W}(A)$. By Proposition 5.2.10 the syntactic Ext-algebra of L is contained in $\mathbf{V}$. Since $\mathbf{V}$ is the pseudo-variety of Ext-algebras generated by all syntactic Ext-algebras of languages of $\mathcal{V}$, there exist an $n \in \mathbb{N}$ and for $i = 1, \ldots, n$ visibly pushdown alphabets A_i and languages $L_i \subseteq {A_i}^{\triangle}$ such that $\mathrm{Ext}(L)$ divides the product

$$R := \prod_{i=1}^{n} \mathrm{Ext}(L_i).$$

It follows that also R recognises L and we denote the morphism recognising L by $\varphi\colon A^{\triangle} \to R$. Denote by $\pi_i\colon R \to \textsc{Ext}(L_i)$ the projection on the ith component and by $\varphi_i = \pi_i \circ \varphi$. Then there exist morphisms $\psi_i\colon A^{\triangle} \to A_i$ such that the diagram



commutes. Observe that since R recognises L, there exists some $K \subseteq R$ such that

$$L = \varphi^{-1}(K) = \bigcup_{x \in K} \varphi^{-1}(x)$$

and letting $x = (x_1, \ldots, x_n)$ we obtain

$$\varphi^{-1}(x) = \bigcap_{i=1}^{n} \varphi_i^{-1}(x_i).$$

From the previous diagram, we get $\varphi_i = (\eta_{L_i} \circ \psi_i)$. We conclude that $L \in \mathcal{V}(A)$: Since $\mathcal{V}(A)$ is closed under Boolean combinations and inverse morphisms, it suffices to prove that $\eta_{L_i}^{-1}(x_i) \in \mathcal{V}(A_i)$, which follows directly from Proposition 5.3.2. ■

5.4 The Free Profinite Ext-algebra

This section is structured along the lines of the book of Jean-Éric Pin, in particular the chapter on profinite words.[1]

In the following, we assume a and c to be call and b and d to be return letters, unless indicated otherwise.

We say that an Ext-algebra R separates two well-matched words $u, v \in A^{\triangle}$, if there is a morphism $h\colon A^{\triangle} \to R$ such that $h(u) \neq h(v)$. For instance, the Ext algebra

$\cdot$	1	0
1	1	0
0	0	0

	1	0
$\text{ext}_{a,b}$	0	1

separates the words ab and $aabb$.

[1] Version November 30, 2016: `https://www.irif.fr/ jep/PDF/MPRI/MPRI.pdf`

For $n, m \in \mathbb{N}$ with $n < m$, the words $a^n b^n$ and $a^m b^m$ can be separated by an EXT-algebra $R = \{0, 1, \ldots, m\}$, where $\mathrm{ext}_{a,b}(i) = \min\{i+1, m\}$. If n and m are not congruent modulo some number $i \leq n$, then the two words can even be separated by an EXT-algebra of size at most i.

It might be tempting to think that any two words $u, v \in A^\triangle$ that can be separated by an EXT-algebra of size n can also be separated by a monoid of size n, since the last few examples satisfy that property. This is not the case. Let a, c be call and b, d be return letters, then

$$a^4b^2c^2d^4 \text{ and } a^2b^2c^2d^2$$

are separated by the algebra

$\cdot$	1	0
1	1	0
0	0	0

	1	0
$\mathrm{ext}_{a,b}, \mathrm{ext}_{c,d}$	1	0
$\mathrm{ext}_{a,d}, \mathrm{ext}_{c,b}$	0	0

since there is only one morphism from $A^\triangle$ to that algebra, which sends the word $a^4b^2c^2d^4$ to 0 and the word $a^2b^2c^2d^2$ to 1.

However, the two words can not be distinguished by a finite monoid of size 2, since the only monoids of size 2 are $\mathbb{Z}_2$ and U_1 and any morphism into these monoids maps them to the same element. In fact, it is possible to construct well-matched words for any natural number n, that can be distinguished by the previous EXT-algebra of size 2, but not by a monoid of size n.

Since any two distinct well-matched words u and v can be separated by the syntactic morphism of $\{u\}$, we obtain:

5.4.1 Lemma

For any $u, v \in A^\triangle$ with $u \neq v$, there exists a finite EXT-algebra that separates u and v.

Let

$$r(u, v) = \min\{|R| \mid R \text{ is a finite EXT-algebra that separates } u \text{ and } v\}.$$

5.4.2 Proposition

The map

$$\begin{aligned} d\colon A^\triangle \times A^\triangle &\to [0, \infty) \\ (u, v) &\mapsto 2^{-r(u,v)} \end{aligned}$$

defines a metric on $A^\triangle$ that satisfies the strong triangle inequality: For all $u, v, x \in A^\triangle$

$$d(u, v) \leq \max\{d(u, x), d(x, v)\}.$$

Proof. That d is positive definite follows directly from Lemma 5.4.1. The symmetry of d is clear, hence the strong triangle inequality remains to be shown.

Assume that R is an EXT-algebra separating u and v. Then R must separate u and x or v and x, which results in $r(u,v) \geq \min\{r(u,x), r(x,v)\}$. And since $d(u,v) = 2^{-r(u,v)}$, we obtain that the strong triangle inequality holds. ∎

The set $A^\triangle$ is hardly interesting as a metric space, since it is discrete. But, it is not a complete space: For instance, for any internal letter c, the sequence $(c^{n!})_{n\in\mathbb{N}}$ is Cauchy, but does not converge.

Its completion, which we denote by $\widehat{A^\triangle}$ has a few notable algebraic properties, such as:

5.4.3 Proposition

The following properties hold for $\widehat{A^\triangle}$:

1. The multiplication $\cdot$ on $A^\triangle$ has a unique and uniformly continuous extension $\widehat{\cdot}$ on $\widehat{A^\triangle}$.
2. For all $u, v \in A^*$ with $uv \in A^\triangle$, the maps $\mathrm{ext}_{u,v}$ have unique uniformly continuous extensions $\widehat{\mathrm{ext}_{u,v}} \colon \widehat{A^\triangle} \to \widehat{A^\triangle}$.

Proof. To show that the concatenation $\cdot \colon A^\triangle \times A^\triangle \to A^\triangle$ sending (u,v) to $u \cdot v$, is uniformly continuous, it suffices to prove $d(uv, u'v') \leq d((u,v),(u',v'))$, where the metric on the right is that of the product. Observe that by the strong triangle inequality, for all $u, v, u'v' \in A^\triangle$

$$d(uv, u'v') \leq \max\{d(uv, uv'), d(uv', u'v')\}$$

and, since an EXT-algebra separating uv and uv' (respectively uv' and $u'v'$) also has to separate v and v' (respectively u and u') we obtain

$$d(uv, u'v') \leq \max\{d(uv, uv'), d(uv', u'v')\} \leq \max\{d(v,v'), d(u,u')\}.$$

Hence the concatenation is uniformly continuous with respect to the product metric on $A^\triangle \times A^\triangle$.

Let $x, y \in A^*$ such that $xy \in A^\triangle$. To see that $\mathrm{ext}_{x,y}$ is uniformly continuous, observe that if an EXT-algebra separates $\mathrm{ext}_{x,y}(u)$ and $\mathrm{ext}_{x,y}(v)$, then it must also separate u and v and hence

$$d(\mathrm{ext}_{x,y}(u), \mathrm{ext}_{x,y}(v)) \leq d(u,v),$$

which implies the uniform continuity of $\mathrm{ext}_{x,y}$.

By Corollary 3.3.10, concatenation and $\mathrm{ext}_{x,y}$ have unique uniformly continuous extensions. ∎

However, apart from the maps $\widehat{\mathrm{ext}_{u,v}}\colon \widehat{A^\triangle} \to \widehat{A^\triangle}$, it is possible to derive further maps from $\widehat{A^\triangle}$ to $\widehat{A^\triangle}$ from elements of $\mathcal{O}(A^\triangle)$ in a more general fashion.

5.4.4 Proposition

Let $(e_n)_{n\in\mathbb{N}}$ be a sequence of elements in $\mathcal{O}(A^\triangle)$, such that $(e_n(x))_{n\in\mathbb{N}}$ is Cauchy for each $x \in A^\triangle$. Then the sequence $(e_n)_{n\in\mathbb{N}}$ uniquely determines a uniformly continuous map $e\colon \widehat{A^\triangle} \to \widehat{A^\triangle}$.

Proof. Since $(e_n(x))_{n\in\mathbb{N}}$ is Cauchy for each $x \in A^\triangle$, the map e sending x to $\lim_{n\to\infty} e_n(x)$ is a well-defined map from $A^\triangle$ to $\widehat{A^\triangle}$. Moreover, it is uniformly continuous, since an EXT-algebra that does not separate two well-matched words x, y does also not separate $e_n(x)$ and $e_n(y)$ for each $n \in \mathbb{N}$. Hence $d(e(x), e(y)) \leq d(x, y)$, which implies uniform continuity and thus there exists a uniformly continuous extension $\widehat{e}\colon \widehat{A^\triangle} \to \widehat{A^\triangle}$. ∎

The space $\widehat{A^\triangle}$ becomes an EXT-algebra, with the uniformly continuous extension $\widehat{\cdot}$ of the multiplication on $A^\triangle$ as multiplication on $\widehat{A^\triangle}$ and the set $\mathcal{O}(\widehat{A^\triangle})$ is the set of all maps $e\colon \widehat{A^\triangle} \to \widehat{A^\triangle}$ obtained in the fashion of Proposition 5.4.4.

Observe that $\mathcal{O}(\widehat{A^\triangle})$ indeed is a monoid, since if the element e (resp. f) in $\mathcal{O}(\widehat{A^\triangle})$ is determined by the sequence $(e_n)_{n\in\mathbb{N}}$ (resp. $(f_n)_{n\in\mathbb{N}}$), then $e \circ f$ is determined by $(e_n \circ f_n)_{n\in\mathbb{N}}$.

By sloppiness, we write $\cdot$ instead of $\widehat{\cdot}$ and $\mathrm{ext}_{a,b}$ instead of $\widehat{\mathrm{ext}_{a,b}}$ and assume that it is understood from the context, if we are referring to the extensions.

5.4.5 Proposition

The space $\widehat{A^\triangle}$ is the free profinite EXT-algebra. That is, any uniformly continuous morphism from $\widehat{A^\triangle}$ is uniquely determined by its values on $\mathrm{ext}_{c,\lambda}$ and $\mathrm{ext}_{a,b}$ for $a \in A_C, b \in A_R$ and $c \in A_I$.

Proof. Since any morphism from $A^\triangle$ is uniquely determined by those values, it suffices to show that any morphism $\varphi\colon A^\triangle \to E$ is uniformly continuous – the completeness of E as a discrete metric space, where

$$d_E(x, y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{otherwise.} \end{cases}$$

ensures the existence of a unique uniformly continuous extension.

Let $u, v \in A^\triangle$, then $d(u, v) < 2^{-|E|}$ implies that u and v cannot be separated by E and thus $d_E(\varphi(u), \varphi(v)) = 0$, which shows that φ is uniformly continuous. By Corollary 3.3.10, there exists a unique uniformly continuous extension $\widehat{\varphi}\colon \widehat{A^\triangle} \to E$.

It remains to be shown, that $\widehat{\varphi}$ is a morphism on $\widehat{A^\triangle}$. To see that, we prove that the set

$$H = \{(u, v) \in \widehat{A^\triangle} \times \widehat{A^\triangle} \mid \widehat{\varphi}(u \cdot v) = \widehat{\varphi}(u) \cdot \widehat{\varphi}(v)\}$$

is equal to $\widehat{A^\triangle} \times \widehat{A^\triangle}$ and that

$$H_{ab} = \{u \in \widehat{A^\triangle} \mid \widehat{\varphi}(\mathrm{ext}_{a,b}(u)) = \mathrm{ext}_{a,b}(\widehat{\varphi}(u))\}$$

is equal to $\widehat{A^\triangle}$. Since $\widehat{\varphi}$ is an extension of φ, which is a morphism, $A^\triangle \times A^\triangle$ is contained in H and $A^\triangle$ is contained in H_{ab} and since $A^\triangle$ is dense in $\widehat{A^\triangle}$, it suffices to show that both H and H_{ab} are closed.

Observe that the equality

$$H_{ab} = \bigcup_{x \in E} \left((\widehat{\varphi} \circ \mathrm{ext}_{a,b})^{-1}(\{x\}) \cap (\mathrm{ext}_{a,b} \circ \widehat{\varphi})^{-1}(\{x\})\right)$$

holds and since each singleton $\{x\}$ is a closed set of E and both $\mathrm{ext}_{a,b}$ and $\widehat{\varphi}$ are uniformly continuous, H_{ab} is a finite union of closed sets, which is again closed.

The argument for H is along the same lines. It suffices to recall that the multiplication is a uniformly continuous function $\widehat{m}\colon \widehat{A^\triangle} \times \widehat{A^\triangle} \to \widehat{A^\triangle}$. Then

$$\widehat{\varphi}(u \cdot v) = (\widehat{\varphi} \circ \widehat{m})(u, v) \text{ and } \widehat{\varphi}(u) \cdot \widehat{\varphi}(v) = (\widehat{m} \circ (\widehat{\varphi} \times \widehat{\varphi}))(u, v)$$

and by a similar argument as before, H is closed. Thus, $\widehat{\varphi}$ is a morphism and $\widehat{A^\triangle}$ is the free profinite Ext-algebra. ∎

Observe that there exist morphisms which are not continuous: The morphism $\varphi\colon \widehat{A^\triangle} \to E$ into an Ext-algebra with U_1 as monoid component, given by

$$\varphi(w) = \begin{cases} 1 & \text{if } w \in A_I^*, \\ 0 & \text{otherwise} \end{cases}$$

is not continuous and does not have a unique extension. It is possible to extend the morphism sending $w \in \widehat{A^\triangle} \backslash A_I^*$ either to 0 or to 1. Both choices contribute non-continuous morphisms. The continuous extension of the restriction of φ to $A^\triangle$ is the morphism equal to the constant morphism 1.

Elements of the Free Profinite Ext-algebra

Imitating the term "profinite words" for elements of the free profinite monoid, we propose to call elements of $\widehat{A^\triangle}$ profinite well-matched words.

There are uncountably many profinite well-matched words, independent of the alphabet we choose, but some elements of $\widehat{A^\triangle}$ have relatively concrete characterisations. For instance, for $x \in \widehat{A^\triangle}$, the sequence $x^{n!}$ is a Cauchy sequence. The proof is exactly

the same as in the profinite case, since it operates entirely on the monoid component of EXT-algebras. Borrowed from profinite words, we use the notation

$$\lim_{n\to\infty} x^{n!} = x^\omega.$$

Also, x^ω is an idempotent with respect to the multiplication on $\widehat{A^\triangle}$. In a similar fashion, we obtain for the $\mathrm{ext}_{a,b}$ operations:

5.4.6 Proposition

For $x \in \widehat{A^\triangle}$, the sequence $\mathrm{ext}_{a,b}^{n!}(x)$ is a Cauchy sequence.

Proof. Let $n, m, N \in N$. We show that for any $n, m \geq N$, the profinite well-matched words $\mathrm{ext}_{a,b}^{n!}(x)$ and $\mathrm{ext}_{a,b}^{m!}(x)$ cannot be separated by an EXT-algebra of size at most N. Assume that $\varphi\colon \widehat{A^\triangle} \to E$ is a morphism, where E is an EXT-algebra with $|E| \leq N$. Since E is finite, there exists an $r \in \mathbb{N}$ such that $\mathrm{ext}_{a,b}^r(x) = \mathrm{ext}_{a,b}^r(\mathrm{ext}_{a,b}^r(x)) = \mathrm{ext}_{a,b}^{2r}(x)$, with $r \leq N$. Since $m, n \geq N$, r divides both $n!$ and $m!$ and hence $\mathrm{ext}_{a,b}^{n!}(x) = \mathrm{ext}_{a,b}^r(x) = \mathrm{ext}_{a,b}^{m!}(x)$. This shows that $\mathrm{ext}_{a,b}^{n!}(x)$ and $\mathrm{ext}_{a,b}^{m!}(x)$ cannot be separated by an EXT-algebra of size at most N and hence the sequence $\mathrm{ext}_{a,b}^{n!}(x)$ is a Cauchy-sequence. ∎

Hence the limit of the sequence $\mathrm{ext}_{a,b}^{n!}(x)$ exists in $\widehat{A^\triangle}$ and we define

$$\lim_{n\to\infty} \mathrm{ext}_{a,b}^{n!}(x) = \mathrm{ext}_{ab}^\omega(x).$$

From a first glance, since the well-matched words are a subset of A^*, it might seem convenient to write $\mathrm{ext}_{a,b}^\omega(x) = a^\omega x b^\omega$. But this notation is *very* misleading for the following reason:

The profinite word a^ω is an idempotent in $\widehat{A^*}$ and thus it holds that

$$a^\omega a^\omega b^\omega c^\omega d^\omega d^\omega = a^\omega b^\omega c^\omega d^\omega$$

in $\widehat{A^*}$. But in $\widehat{A^\triangle}$

$$\mathrm{ext}_{ad}^\omega(\mathrm{ext}_{ab}^\omega(\lambda)\mathrm{ext}_{cd}^\omega(\lambda)) \neq \mathrm{ext}_{ab}^\omega(\lambda)\mathrm{ext}_{cd}^\omega(\lambda),$$

since a morphism into the EXT-algebra

$\cdot$	1	ab	cd	$abcd$
1	1	ab	cd	$abcd$
ab	ab	0	$abcd$	0
cd	cd	0	0	0
$abcd$	$abcd$	0	0	0

	1	ab	cd	$abcd$
$\mathrm{ext}_{a,b}$	ab	ab	0	0
$\mathrm{ext}_{c,d}$	cd	0	cd	0
$\mathrm{ext}_{a,d}$	0	0	0	0
$\mathrm{ext}_{c,b}$	0	0	0	0

sends the left side of the equation to 0 and the right side to $abcd$.

In fact, $\mathrm{ext}^{\omega}_{ab}(x)$ has little to do with idempotents of words, but should be thought of as an idempotent in the monoid that is generated by the maps $\mathrm{ext}^{\omega}_{ab}$ with composition as multiplication.

It is straight-forward to see, that $d(\mathrm{ext}^{\omega}_{ab}(x), \mathrm{ext}^{\omega}_{ab}(y)) \leq d(x, y)$ and it thus holds that:

5.4.7 Proposition

The map $\mathrm{ext}^{\omega}_{a,b}$ with $x \mapsto \mathrm{ext}^{\omega}_{ab}(x)$ is uniformly continuous.

Then, we obtain that for any morphism $h\colon \widehat{A^{\triangle}} \to E$ into a finite Ext-algebra E,

$$h(\mathrm{ext}^{\omega}_{ab}(x)) = \mathrm{ext}^{\omega}_{ab}(h(x))$$

where $\mathrm{ext}^{\omega}_{ab}$ is indeed the idempotent generated by $\mathrm{ext}_{a,b}$ in the monoid of maps from E to E. This also implies $\mathrm{ext}^{\omega}_{ab} \circ \mathrm{ext}^{\omega}_{ab} = \mathrm{ext}^{\omega}_{ab}$.

One may also prove that the sequence of maps $(\mathrm{ext}^{\omega}_{ab})^{n!}$ converges to $\mathrm{ext}^{\omega}_{ab}$, when the set of all maps from $\widehat{A^{\triangle}}$ to $\widehat{A^{\triangle}}$ is identified with the product of $\widehat{A^{\triangle}}$ and equipped with the product topology.

The Stone Space of the Visibly Pushdown Languages

While Stone spaces in general are not necessarily equipped with algebraic structure, such as a multiplication, we will now show that the free profinite Ext-algebra is isomorphic to the Stone space of the visibly pushdown languages over A.

5.4.8 Proposition

The space $\widehat{A^{\triangle}}$ is compact.

Proof. For $u, v \in \widehat{A^{\triangle}}$, define the relation $\sim_n$ by $u \sim_n v$, if u and v cannot be separated by an Ext-algebra of size at most n. It is evident that $u \sim_n v$ is reflexive and symmetrical. Let $x \in A^{\triangle}$ with $u \sim_n x$ and $x \sim_n v$. Since any monoid separating u and v must separate u and x or v and x, transitivity follows and $\sim_n$ is an equivalence relation. More precisely, it is also of finite index, since there are only finitely many Ext algebras of size at most n. Since, the equivalence class of u is precisely the open ball $B_{2^{-n}}(u)$ on $A^{\triangle}$, it is covered by finitely many open balls of radius 2^{-n}. Since the open balls form a basis of the topology, $\widehat{A^{\triangle}}$ is compact. ∎

In the same way in which we defined the syntactic Ext-algebra of a language $L \subseteq A^{\triangle}$ we define the syntactic Ext-algebra of a subset $R \subseteq A^{\triangle}$, that is the quotient induced by the congruence on $\widehat{A^{\triangle}}$ given by $u \sim_R v$ if for all $e \in \mathcal{O}(\widehat{A^{\triangle}})$ it holds that $u \in e^{-1}(R) \Leftrightarrow v \in e^{-1}(R)$.

5.4.9 Proposition

Let $R \subseteq \widehat{A^\triangle}$ and let $\textsc{Ext}(R)$ be its syntactic $\textsc{Ext}$-algebra. Then the following conditions are equivalent

1. R is clopen,
2. the syntactic congruence of R is a clopen subset of $\widehat{A^\triangle} \times \widehat{A^\triangle}$
3. $\textsc{Ext}(R)$ is finite and the syntactic morphism of R is a continuous map.

Proof. **1. to 2.:** Let R be a clopen subset of $\widehat{A^\triangle}$. Define the set

$$X = \bigcap_{e \in \mathcal{O}(\widehat{A^\triangle})} ((e^{-1}(R) \times e^{-1}(R)) \cup (e^{-1}(R^c) \times e^{-1}(R^c)))$$

Then X is the syntactic congruence $\sim_R$ of R.

We prove that X is closed. Observe that since R is clopen and by definition of $\mathcal{O}(\widehat{A^\triangle})$, each e is a continuous map, each of the sets $e^{-1}(R)$ is closed and so is $e^{-1}(R^c)$. Since $\widehat{A^\triangle} \times \widehat{A^\triangle}$ has the product topology and arbitrary intersections of closed sets are closed, X is closed.

We prove that $\sim_R^c$ is closed, by showing that the limit of any convergent sequence in $\sim_R^c$ is contained in it. Let $((s_n, t_n))_{n\in\mathbb{N}}$ be a convergent sequence in $\sim_R^c$ with limit (s, t). Since s_n and t_n are not equivalent, there exists a sequence $e_n \in \mathcal{O}(\widehat{A^\triangle})$ such that, without loss of generality, $e_n(s_n) \in R$ and $e_n(t_n) \notin R$.

Because $\widehat{A^\triangle}$ is compact, so is its product $\widehat{A^\triangle} \times \widehat{A^\triangle}$ and hence $(e_n)_{n\in\mathbb{N}}$ has a convergent subsequence $(e_{i_j})_{j\in\mathbb{N}}$ with limit $e \in \mathcal{O}(\widehat{A^\triangle})$. Since R is clopen and each e_{i_j} is continuous, $e_{i_j}(s_n)$ converges to $e(s) \in R$ and $e_{i_j}(t_n)$ converges to $e(t) \notin R$. Thus $(s, t) \in \sim_R^c$, which shows that it is closed. Thus $\sim_R$ is open.

2. to 3.: We show that for any $x \in \widehat{A^\triangle}$, the equivalence class of x with respect to $\sim_R$ is open. It holds that for each $x \in A^\triangle$, $(x, x) \in \sim_R$. Since $\sim_R$ is open, there exists an open set $U \subseteq \widehat{A^\triangle}$ such that $(x, x) \in U \times U \subset \sim_R$. Moreover U must be fully contained in the equivalence class of x, since otherwise there exist two non-equivalent elements $u, v \in U$ with $(u, v) \in U \times U$, which is a contradiction to $U \times U \subset \sim_R$. Since x was arbitrary, this implies that the x-classes of $\sim_R$ are open and thus form an open partition of $\widehat{A^\triangle}$.

By compactness of $\widehat{A^\triangle}$ this partition is finite, which implies that $\textsc{Ext}(R)$ is finite and the syntactic morphism is continuous by the observation that each x-class is the preimage of a singleton in $\textsc{Ext}(R)$ and open.

3. to 1.: Let $\eta_R \colon \widehat{A^\triangle} \to \textsc{Ext}(R)$ be the syntactic morphism of R. Since $\textsc{Ext}(R)$ is finite, it is discrete and each subset of $\textsc{Ext}(R)$ is clopen and since $\eta_R^{-1}(\eta_R(R)) = R$, and η_R is continuous, R is clopen. ∎

5.4.10 Proposition

If $L \subseteq A^\triangle$ is a language, then $L = \overline{L} \cap A^\triangle$ and the following conditions are equivalent:

1. L is a VPL
2. $L = K \cap A^\triangle$ for some clopen set $K \subseteq \widehat{A^\triangle}$
3. $\overline{L}$ is clopen in $\widehat{A^\triangle}$
4. $\overline{L}$ is recognised by a continuous morphism $\varphi\colon \widehat{A^\triangle} \to E$, where E is a finite Ext-algebra.

Proof. **1. to 2.:** Let L be a VPL, then by Proposition 5.2.13, there exists a finite Ext-algebra recognising L. Let $\varphi\colon A^\triangle \to E$ be the morphism recognising L, that is $L = \varphi^{-1}(\varphi(L))$. Moreover, let $K = \widehat{\varphi}^{-1}(\varphi(L))$. Since E is discrete, $\varphi(L)$ is a clopen set and since $\widehat{\varphi}$ is continuous, K is clopen. Then $\varphi(w) = \widehat{\varphi}(w)$ for $w \in A^\triangle$, implies $L = \widehat{\varphi}^{-1}(\varphi(L)) \cap A^\triangle = K \cap A^\triangle$.

2. to 3.: Assume that $L = K \cap A^\triangle$ for some clopen set $K \subseteq \widehat{A^\triangle}$. Since $A^\triangle$ is dense in $\widehat{A^\triangle}$, and since K is open, it follows that $K \cap A^\triangle$ is dense in K. Thus $\overline{K \cap A^\triangle} = K = \overline{L}$ is clopen.

3. to 4.: See Proposition 5.4.9

4. to 1.: Assume that $\overline{L}$ is recognised by a morphism $\varphi\colon \widehat{A^\triangle} \to E$ into a finite Ext-algebra E. Then $\overline{L} = \varphi^{-1}(R)$ for some $R \subseteq E$. Let $\varphi_{A^\triangle}\colon A^\triangle \to E$ be the restriction of φ to $A^\triangle$. Then

$$L = \overline{L} \cap A^\triangle = \varphi^{-1}(R) \cap A^\triangle = \varphi_{A^\triangle}^{-1}(R)$$

and by Proposition 5.2.13, L is a VPL. ∎

The statements of the next Proposition characterise the topological closures of VPLs. This Proposition in particular proves that algebra and topology are extremely closely related on the free profinite Ext-algebra – the Stone spaces of the VPLs.

5.4.11 Proposition

Let $L \subseteq A^\triangle$ be a VPL and let $u \in \widehat{A^\triangle}$. Then the following conditions are equivalent:

1. $u \in \overline{L}$
2. $\widehat{\varphi}(u) \in \varphi(L)$, for all morphisms φ from $A^\triangle$ into a finite Ext-algebra.
3. $\widehat{\eta}(u) \in \eta(L)$, where η is the syntactic morphism of L.
4. $\widehat{\varphi}(u) \in \varphi(L)$, for some morphism φ from $A^\triangle$ into a finite Ext-algebra recognising L.

Proof. **1. to 2.:** Suppose $u \in \overline{L}$ and let $\varphi\colon A^\triangle \to R$ be a morphism into a finite Ext-algebra R. Since $\widehat{\varphi}$ is continuous and R is discrete we have

$$\widehat{\varphi}(\overline{L}) = \overline{\widehat{\varphi}(L)} = \widehat{\varphi}(L) = \varphi(L).$$

2. to 3. and 3. to 4.: Is trivial.

4. to 1.: Let $\varphi\colon A^\triangle \to R$ be a morphism into a finite Ext-algebra R. If R recognises L, then $L = \varphi^{-1}(\varphi(L))$, which together with continuity of $\widehat{\varphi}$ and discreteness of R implies

$$\overline{L} = \overline{\varphi^{-1}(\varphi(L))} = \widehat{\varphi}^{-1}(\widehat{\varphi}(\overline{L})) = \widehat{\varphi}^{-1}(\varphi(L)).$$

■

The following theorem is a conclusion from the previous propositions, in particular 5.4.10.

Denote by $\mathrm{VPL}(A)$ the set of visibly pushdown languages over the visibly pushdown alphabet A and by $\mathrm{Clopen}(\widehat{A^\triangle})$ the set of all clopen sets of $\widehat{A^\triangle}$.

5.4.12 Theorem

The maps

$$\begin{aligned} \mathrm{VPL}(A) &\to \mathrm{Clopen}(\widehat{A^\triangle}) \\ L &\mapsto \overline{L} \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mathrm{Clopen}(\widehat{A^\triangle}) &\to \mathrm{VPL}(A) \\ K &\mapsto A^\triangle \cap K \end{aligned}$$

are morphisms of Boolean algebras and inverse to each other.

Proof. That both maps are inverse to each other, follows directly from Proposition 5.4.10. It remains to be shown that both are morphisms of Boolean algebras, which is straight-forward for the map $K \mapsto A^\triangle \cap K$. That $L \mapsto \overline{L}$ is a morphism can be derived from the observation that $\overline{L} = \widehat{\eta}^{-1}(\eta(L))$, where η is the syntactic morphism of L and the fact that closure and union commute. ■

5.5 A Reiterman Theorem for VPL

Pseudo-varieties of VPL can, just like pseudo-varieties of regular languages, be characterised through sets of equations which are now tuples of elements of the free profinite Ext-algebra. We introduce the appropriate notions and prove a Reiterman-like theorem.

Let $u, v \in \widehat{A^\triangle}$ be two profinite well-matched words. We say that a VPL L satisfies the equation $[u \leftrightarrow v]$ if and only if

$$u \in \overline{L} \Leftrightarrow v \in \overline{L}$$

and similarly we say that an EXT-algebra R satisfies the equation $[u \leftrightarrow v]$ if and only if for each morphism $\varphi\colon A^{\triangle} \to R$

$$\widehat{\varphi}(u) = \widehat{\varphi}(v).$$

The following Corollary is a direct consequence of Proposition 5.4.11.

5.5.1 Corollary

Let $L \subseteq A^{\triangle}$ be a VPL and let $u, v \in \widehat{A^{\triangle}}$, then the following statements are equivalent:

1. L satisfies the equation $[u \leftrightarrow v]$.
2. the syntactic EXT-algebra of L satisfies $[u \leftrightarrow v]$.
3. each EXT-algebra recognising L satisfies $[u \leftrightarrow v]$.

Let E be a set of equations over $\widehat{A^{\triangle}}$. We define the set $[\![E]\!]$ to be the set of all EXT-algebras satisfying all the equations in E and the set $L([\![E]\!])$ all languages satisfying all equations in E. From the previous Corollary, we obtain that the languages recognised by $[\![E]\!]$ are precisely the languages in $L([\![E]\!])$, which justifies the notation.

5.5.2 Proposition

If E is a set of equations over $\widehat{A^{\triangle}}$, then $[\![E]\!]$ is a pseudo-variety of EXT-algebras and $L([\![E]\!])$ is the corresponding pseudo-variety of languages.

Proof. First of all, observe that the intersection of pseudo-varieties is a pseudo-variety and that thus, without loss of generality, it suffices to show that $[\![u \leftrightarrow v]\!]$ for some $u, v \in \widehat{A^{\triangle}}$ forms a variety.

In the following, we omit the details since they are utterly straight-forward once written down. That quotients and direct products preserve equations, results in $[\![u \leftrightarrow v]\!]$ being a pseudo-variety of EXT-algebras and by the observation that $L([\![u \leftrightarrow v]\!])$ are the languages recognised by $[\![u \leftrightarrow v]\!]$ the claim follows. ∎

5.5.3 Theorem

A class of EXT-algebras (resp. of VPL) is a pseudo-variety if and only if it can be defined by a set of identities.

Proof. One direction of the proof follows directly from Proposition 5.5.2. For the other direction, let $\mathbf{V}$ be a pseudo-variety of EXT-algebras and let E be the set of equations that are satisfied by all elements of $\mathbf{V}$. Now, it is evident that $\mathbf{V} \subseteq [\![E]\!]$.

We prove that the inclusion $[\![E]\!] \subseteq \mathbf{V}$ holds. Let $R \in [\![E]\!]$ and $\varphi\colon \widehat{A^\triangle} \to R$ be a morphism. For an element $S \in \mathbf{V}$ and a morphism $h\colon A^\triangle \to S$, we define the set

$$N_h = \{(u,v) \in \widehat{A^\triangle} \times \widehat{A^\triangle} \mid \widehat{h}(u) \neq \widehat{h}(v)\}.$$

This set is open, since $\widehat{h}$ is continuous and N_h is the preimage of the complement of the diagonal of $S \times S$, which is open. Formally, we also identify E with a subset of $\widehat{A^\triangle} \times \widehat{A^\triangle}$, that is the set of all (u,v) such that $[u \leftrightarrow v]$ is an equation in E. We observe that if a tuple (u,v) is not contained in any set N_h for some morphism h, then (u,v) must be in E. Denote by $\mathcal{F}$ the set of all morphisms from $A^\triangle$ into an Ext-algebra of $\mathbf{V}$. It follows that

$$E \cup \bigcup_{h \in \mathcal{F}} N_h$$

is a cover of $\widehat{A^\triangle} \times \widehat{A^\triangle}$. Define the set

$$E_\varphi = \{(u,v) \in \widehat{A^\triangle} \times \widehat{A^\triangle} \mid \widehat{\varphi}(u) = \widehat{\varphi}(v)\}.$$

Clearly $E \subseteq E_\varphi$ and E_φ is open. By the previous argumentation,

$$E_\varphi \cup \bigcup_{h \in \mathcal{F}} N_h$$

is an open cover of $\widehat{A^\triangle} \times \widehat{A^\triangle}$ an hence there exists a finite subcover for some finite subset $F \subseteq \mathcal{F}$

$$E_\varphi \cup \bigcup_{h \in F} N_h.$$

In particular, if $\widehat{h}(u) = \widehat{h}(v)$ for all $h \in F$, then also $(u,v) \notin N_h$ for all $h \in F$ and hence $(u,v) \in E_\varphi$, which implies $\widehat{\varphi}(u) = \widehat{\varphi}(v)$. Hence $\widehat{\varphi}$ factors through the product of all $\widehat{h}$, that is the morphism sending $u \in \widehat{A^\triangle}$ to $(\widehat{h}(u))_{h \in F}$. This implies that R is a quotient of the product $\prod_{h \in F} \widehat{h}(\widehat{A^\triangle})$ and since the morphisms $h \in F$ have elements of $\mathbf{V}$ as co-domains and $\mathbf{V}$ is closed under subs and finite products, $\prod_{h \in F} \widehat{h}(\widehat{A^\triangle})$ is an element of $\mathbf{V}$. Thus, R is also in $\mathbf{V}$. ∎

5.6 Concepts in Application

This section serves not only as a proof of concept for the developed theory, but also to investigate equations for two subclasses of visibly pushdown languages: The set of languages of the form $A^\triangle \cap K$, where K is a regular language and, since these languages are precisely the visibly counter languages with threshold zero, subsequently also the visibly counter languages (with arbitrary threshold). We start by approaching the former set of languages from an algebraic point of view.

Monoidal Ext-algebras

Let A be a visibly pushdown alphabet and let $h\colon A^* \to M$ be a morphism into a finite monoid M. Then the set $R := h(A^{\triangle}) \subseteq M$ together with the multiplication from M and the generating operations

$$\begin{aligned} \mathrm{ext}_{a,b}\colon R &\to R \\ x &\mapsto h(a) \cdot x \cdot h(b) \end{aligned}$$

is an Ext-algebra. It is clear that since M is finite, also R is finite and hence only recognises VPL, but the finiteness of M is just a sufficient and not a necessary condition for the finiteness of R.

For instance, let M be the syntactic monoid of the language $\{a^n b^n \mid n \in \mathbb{N}\}$ and let $\eta\colon \{a,b\}^* \to M$ be its syntactic morphism. Then the Ext-algebra generated in the way described above also is finite, in fact $R = \{1, ab, 0\}$, where 1 is the equivalence class of the empty word and 0 the equivalence class of $abab$. This is precisely the syntactic Ext-algebra of $\{a^n b^n \mid n \in \mathbb{N}\}$.

In fact, construction is possible for every VPL and it follows, that the proposition

5.6.1 Proposition

Let L be a VPL. Then the syntactic Ext-algebra of L is isomorphic to the Ext-algebra generated by the syntactic (monoid) morphism of L.

holds. Since the proof is straight-forward, we omit it.

A more interesting question to ask is: Which Ext-algebras can be obtained from *finite* monoids? For instance, the syntactic monoid of a^*b^* generates the syntactic Ext-algebra of $\{a^n b^n \mid n \in \mathbb{N}\}$. We try to approach this question through the developed topological and algebraic methods. The decidability of the question was shown in [BLS06] via visibly counter automata.

5.6.2 Definition

We say that a finite Ext-algebra R is monoidal, if there exists a morphism $h\colon A^* \to M$ into a finite monoid M, such that R is isomorphic to the Ext-algebra $h(A^{\triangle})$ with the generators of operations $\mathrm{ext}_{a,b}(x) = h(a) \cdot x \cdot h(b)$ and left- and right multiplication.

Observe that in the following, we omit to mention the left and right multiplication $x \mapsto r \cdot x$ and $x \mapsto x \cdot r$, for $x, r \in R$ when we say the operations are generated by $\mathrm{ext}_{a,b}(x) = h(a) \cdot x \cdot h(b)$, since we assume that it is understood they are included.

For instance, the syntactic Ext-algebras of the languages $\{a^{2n} b^{2n} \mid n \in \mathbb{N}\}$ or $(ab)^*$ are monoidal. Equivalently, an Ext-algebra is monoidal, if it is isomorphic to an

EXT-algebra that is the submonoid of some finite monoid M with operations given by $e(x) = x_a \cdot x \cdot x_b$ for some elements $x_a, x_b \in M$.

By **M**EXT, we denote the class of all monoidal EXT-algebras.

5.6.3 Proposition

The class **M**EXT is a pseudo-variety.

Proof. We prove that **M**EXT is closed under quotients, taking subalgebras and finite direct products.

Let $R \in$ **M**EXT and let M be the monoid and $h\colon A^* \to M$ be the morphism generating R. Without loss of generality, we assume that h is surjective, since otherwise we restrict to the image of h. Then R is generated by the set $h(A_I)$ and the monoid of operations $\mathcal{O}(R)$ is generated by $\mathrm{ext}_{a,b}$ for $a \in A_C$ and $b \in A_R$.

Let S be a quotient of R. Then there exists a surjective morphism (of EXT-algebras) $\varphi\colon R \to S$.

Define the relation

$$\sim_\varphi := \{(h(u), h(v)) \mid u, v \in A^\triangle \text{ with } \varphi(h(u)) = \varphi(h(v))\}$$

on M and let $\equiv_\varphi$ be the congruence relation on M generated by $\sim_\varphi$. Define the monoid $N := M\backslash_{\equiv_\varphi}$ and let $\psi\colon M \to N$ be the canonical morphism induced by the congruence. It follows from the finiteness of M that also N is finite. Then φ and ψ coincide by definition on images of well-matched words and since ψ is a morphism of monoids, for any $x \in M$ the equality

$$\varphi(\mathrm{ext}_{a,b})(\varphi(x)) = \varphi(\mathrm{ext}_{a,b}(x)) = \varphi(h(a) \cdot x \cdot h(b)) = \psi(h(a)) \cdot x \cdot \psi(h(b))$$

holds. Since moreover φ is surjective, S is generated by $\psi \circ h\colon A^* \to N$ and thus **M**EXT is closed under quotients.

Let S be a subalgebra of R. Then there exists an $n \leq |A_I|$ and words $w_1, \ldots, w_n \in A^\triangle$ and a $k \leq |A_C| = |A_R|$ and words $u_1, \ldots, u_k \in A^*$, $v_1, \ldots, v_k \in A^*$ with $u_i v_i \in A^\triangle$ for $i = 1, \ldots, k$ such that S is generated by $h(w_1), \ldots, h(w_n)$ and the monoid of operations on S by ext_{u_i,v_i}. Choose some enumeration of the call-, return- and internal letters and define the morphism of monoids $g\colon A^* \to M$ by sending the ith letter of A_C (resp. A_R, A_I) to $h(u_i)$ (resp. $h(v_i)$, $h(w_i)$), if i does not exceed k (resp. n) and to the neutral element of M otherwise. By construction, g generates S and hence $S \in$ **M**EXT.

The closure under product is clear: If S and R are monoidal, then $S \times R$ is generated by the product-morphism generating S and R. ∎

By $\mathcal{M}$EXT denote the corresponding pseudo-variety of VPL. It follows immediately that a language belongs to $\mathcal{M}$EXT if and only if its syntactic EXT-algebra belongs to **M**EXT.

5.6.4 Proposition

Let $L \subseteq A^{\triangle}$ be a language. Then L belongs to $\mathcal{M}\textsc{Ext}$ if and only if there exists a regular language K such that

$$L = A^{\triangle} \cap K.$$

Proof. Suppose that $L \subseteq A^{\triangle}$ is in $\mathcal{M}\textsc{Ext}$ and let $\textsc{Ext}(L)$ be its syntactic Ext-algebra and η_L its syntactic morphism. Then there exists a finite monoid M such that $\textsc{Ext}(L)$ is isomorphic to a submonoid of M with $\mathrm{ext}_{a,b}(x) = x_a \cdot x \cdot x_b$ for some $x_a, x_b \in M$. Define the morphism $h\colon A^* \to M$ by sending any internal letter c to $\eta_L(c)$, a call letter a to its respective element x_a and similarly $b \in A_R$ to x_b. Define $K = h^{-1}(h(L))$. Since M is finite, K is a regular languages and it follows from the definition of h, that it coincides with η_L on well-matched words, which proves one direction of the claim.

Let K be a regular language and let $L = A^{\triangle} \cap K$. By M_K denote the syntactic monoid of K and by $\eta_K\colon A^* \to M_K$ its syntactic morphism. Then η_K generates an Ext-algebra $R \subseteq M_K$ and the morphism $\varphi\colon A^{\triangle} \to R$ induced by η_K recognises $L = \varphi^{-1}(\varphi(L))$. Since M_K is finite, it follows that $L \in \mathcal{M}\textsc{Ext}$. ∎

A consequence of these considerations, which is not surprising is

5.6.5 Corollary

If K is a regular language over A^*, then $A^{\triangle} \cap K$ is visibly pushdown.

Equations

Having made sure that $\mathbf{M}\textsc{Ext}$ is a pseudo-variety, we know that it is characterised by a set of equations over well-matched words. We are now going to derive a set of equations which is sound in the sense that every language in $\mathcal{M}\textsc{Ext}$ satisfies the equations. Since we are dealing with VPL inherently connected to monoids, we begin by drawing a connection between the free profinite monoid over A and the free profinite Ext-algebra.

5.6.6 Proposition

The inclusion $\iota\colon A^{\triangle} \to A^*$ is uniformly continuous.

Proof. Let $u, v \in A^{\triangle}$. We show that $d(\iota(u), \iota(v)) \leq d(u, v)$. Assume that u and v are separated by a monoid M. Then, there exists a morphism $h\colon A^* \to M$ such that $h(u) \neq h(v)$ and the Ext-algebra $h(A^{\triangle})$ with the multiplication of M and $\mathrm{ext}_{a,b}(x) = h(a) \cdot x \cdot h(b)$ separates u and v. Since $|h(A^{\triangle})| \leq |M|$, we obtain $r(u, v) \leq r(\iota(u), \iota(v))$ and conclude $d(\iota(u), \iota(v)) \leq d(u, v)$, which proves the claim. ∎

It follows that ι has a unique uniformly continuous extension $\widehat{\iota}$. For instance, the profinite well-matched word $\mathrm{ext}^{\omega}_{ab}(w)$ maps under $\widehat{\iota}$ to $a^{\omega}wb^{\omega}$. This observation can be connected with equations in the following way:

5.6.7 Proposition

Let $u, v \in \widehat{A^{\triangle}}$. Then **M**Ext satisfies the equation $[u \leftrightarrow v]$ if and only if $\widehat{\iota}(u) = \widehat{\iota}(v)$.

Proof. The direction stating that **M**Ext satisfies $[u \leftrightarrow v]$ implies that $\widehat{\iota}(u) = \widehat{\iota}(v)$ follows directly from the uniform continuity of ι.

For the converse direction, let M be a finite monoid and $R \in \mathbf{M}\mathrm{Ext}$, such that R is isomorphic to some submonoid of M and the operations on R can be represented as usual by multiplication in M. Now it is clear, that if $\widehat{\iota}(u) = \widehat{\iota}(v)$, then $\widehat{h}(\widehat{\iota}(u)) = \widehat{h}(\widehat{\iota}(v))$ for any morphism $h\colon A^* \to M$. Since any morphism $g\colon A^{\triangle} \to R$ is of the form $h \circ \iota$ for some morphism h and hence $\widehat{g} = \widehat{h} \circ \widehat{\iota}$, we obtain $\widehat{g}(u) = \widehat{g}(v)$. ∎

We are now ready to state some sound equations for the pseudo-variety of monoidal Ext-algebras. Observe that equations are transitive in the sense that if a language satisfies $[u \leftrightarrow v]$ and $[v \leftrightarrow w]$ then it satisfies $[u \leftrightarrow w]$. We thus use the short notation $[u \leftrightarrow v \leftrightarrow w]$.

5.6.8 Proposition

For each $x, y, z \in \widehat{A^{\triangle}}$ and $u, v, u', v' \in A^*$ such that $uv, uv', u'v, u'v' \in A^{\triangle}$, the variety **M**Ext satisfies the equations

$$[\mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u,v'}(x) \cdot y \cdot \mathrm{ext}^{\omega}_{u',v}(z)) \leftrightarrow \mathrm{ext}^{\omega}_{u.v'}(x) \cdot y \cdot \mathrm{ext}^{\omega}_{u',v}(z)] \tag{5.1}$$

$$[\mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u',v'}(x)) \leftrightarrow \mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u,v'}(\mathrm{ext}^{\omega}_{u',v'}(x))) \tag{5.2}$$

$$\leftrightarrow \mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u',v}(\mathrm{ext}^{\omega}_{u',v'}(x)))] \tag{5.3}$$

Proof. By Proposition 5.6.7, **M**Ext satisfies the equation

$$[\mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u,v'}(x) \cdot y \cdot \mathrm{ext}^{\omega}_{u',v}(z)) \leftrightarrow \mathrm{ext}^{\omega}_{u,v'}(x) \cdot y \cdot \mathrm{ext}^{\omega}_{u',v}(z)].$$

if and only if

$$u^{\omega}u^{\omega}xv'^{\omega}yu'^{\omega}uv^{\omega}v^{\omega} = u^{\omega}xv'^{\omega}yu'^{\omega}zv^{\omega}.$$

Since u^{ω} is an idempotent in $\widehat{A^*}$ and hence $u^{\omega}u^{\omega} = u^{\omega}$, the equality holds.

Similarly

$$u^{\omega}u^{\omega}u'^{\omega}v'^{\omega}v'^{\omega}v^{\omega} = u^{\omega}u'^{\omega}v'^{\omega}v^{\omega} = u^{\omega}u'^{\omega}u'^{\omega}v'^{\omega}v^{\omega}v^{\omega}$$

which proves the validity of equations 5.2 and 5.3 for **M**Ext. ∎

Application of Equations for Concrete Languages

In the following, we use the equations for **M**Ext to prove that some VPL are not monoidal. For instance, a simple example, which does not satisfy the first equation, is the language

$$\{a^n b^n c^m d^m \mid n, m \in \mathbb{N}\}$$

over the alphabet $A = \{a, b, c, d\}$ where a, c are call and b, d are return letters. The syntactic Ext-algebra of that language is displayed in Figure 5.3. There we obtain

$$0 = \mathrm{ext}^{\omega}_{a,d}(\mathrm{ext}^{\omega}_{a,b}(1) \cdot \mathrm{ext}^{\omega}_{c,d}(1)) \neq \mathrm{ext}^{\omega}_{a,b}(1) \cdot \mathrm{ext}^{\omega}_{c,d}(1) = abcd.$$

$\cdot$	1	ab	cd	$abcd$
1	1	ab	cd	$abcd$
ab	ab	$abcd$	0	0
cd	cd	0	0	0
$abcd$	$abcd$	0	0	0

$\cdot$	1	ab	cd	$abcd$
$\mathrm{ext}_{a,b}$	ab	ab	0	0
$\mathrm{ext}_{a,d}$	0	0	0	0
$\mathrm{ext}_{c,b}$	0	0	0	0
$\mathrm{ext}_{c,d}$	cd	0	cd	0

Figure 5.3: Syntactic Ext-algebra of the language $\{a^n b^n c^m d^m \mid n, m \in \mathbb{N}\}$.

The second equation is not satisfied by the Ludwig language over the alphabet $A = \{a, b, c\}$ where a is a call, b a return and c an internal letter, which we recall is given by the production rules

$$S \to aScb \mid acSb \mid \lambda.$$

Also recall that its syntactic Ext-algebra is the one displayed in Figure 5.4 We

$\cdot$	1	c	acb	$acbc$	0
1	1	c	acb	$acbc$	0
c	c	0	$acbc$	0	0
acb	acb	$acbc$	0	0	0
$acbc$	$acbc$	0	0	0	0
0	0	0	0	0	0

	1	c	acb	$acbc$	0
$\mathrm{ext}_{a,b}$	0	acb	0	acb	0
$\mathrm{ext}_{\lambda,c}$	c	0	acbc	0	0
$\mathrm{ext}_{a,cb}$	acb	0	acb	0	0
$\mathrm{ext}_{ac,b}$	acb	0	acb	0	0
$\mathrm{ext}_{ac,cb}$	0	0	0	0	0

Figure 5.4: Syntactic Ext-algebra of the Ludwig language.

obtain that

$$acb = \mathrm{ext}^{\omega}_{ac,b}(\mathrm{ext}^{\omega}_{a,cb}(1)) \neq \mathrm{ext}^{\omega}_{ac,b}(\mathrm{ext}^{\omega}_{ac,cb}(\mathrm{ext}^{\omega}_{a,cb}(1))) = 0.$$

As a last example, consider the language H^+ over $A = \{a, b\}$ which was given by the production rules

$$S \to aNb \mid SS \mid \lambda,\ N \to aSb \mid NN \mid NS \mid SN,$$

where a is a call and b a return letter.

It is an instance of a language, where it is less trivial to find representatives for the Ext-operations in order to see that an equation is violated.

Then, H^+ is recognised by the Ext-algebra

$\cdot$	1	0
1	1	0
0	0	0

	1	0
$\mathrm{ext}_{a,b}$	0	1
$^*\mathrm{ext}_{a^2,b^2}$	1	0
$^*\mathrm{ext}_{a^2b,b}$	1	1
$^*\mathrm{ext}_{a^2ab,b^2}$	0	0

Figure 5.5: The syntactic Ext-algebra of H^+ with complete monoid of operations. The idempotent operations are marked by stars.

Evidently $\mathrm{ext}_{u,v} = \mathrm{ext}_{a^2,b^2}$ is the identity, since it corresponds to double negation, and as such is idempotent. Hence $\mathrm{ext}_{u,v} = \mathrm{ext}^\omega_{u,v}$. Moreover, $\mathrm{ext}_{u',v} = \mathrm{ext}^2_{a^2b,b}$ and $\mathrm{ext}_{a^2b,b}$ is the constant map 1, since multiplication by ab corresponds to conjunction with `false` and $\mathrm{ext}_{a,b}$ corresponds to negation. In particular $\mathrm{ext}_{u',v}$ is also idempotent, which results in $\mathrm{ext}_{u',v} = \mathrm{ext}^\omega_{u',v}$. By a similar argument $\mathrm{ext}_{u,v'} = \mathrm{ext}_{a^2,abb^2}$ is the constant map 0. We obtain that thus independently of the choice of x:

$$0 = \mathrm{ext}^\omega_{u,v}(\mathrm{ext}^\omega_{u',v}(\mathrm{ext}^\omega_{u',v'}(x))) \neq \mathrm{ext}^\omega_{u,v}(\mathrm{ext}^\omega_{u,v'}(\mathrm{ext}^\omega_{u',v'}(x))) = 1$$

or more explicitly

$$\mathrm{ext}^\omega_{a^2,b^2}(\mathrm{ext}^\omega_{a^2ba^2b,b^2}(\mathrm{ext}^\omega_{a^2ba^2b,abb^2}(x))) \neq \mathrm{ext}^\omega_{a^2,b^2}(\mathrm{ext}^\omega_{a^2,abb^2}(\mathrm{ext}^\omega_{a^2ba^2b,abb^2}(x))).$$

Thus, neither the Ludwig Language not H^+ are in $\mathcal{M}$Ext. We prove now, that the equation both languages violate even implies a stronger statement: Both languages are not visibly counter languages (VCL). The language $\{a^nb^nc^md^m \mid n, m \in \mathbb{N}\}$ however is a VCL.

Visibly counter languages

We already mentioned that the decidability of membership to $\mathcal{M}$Ext was shown in [BLS06] via visibly counter automata. In fact, equations 5.2 and 5.3 relate to visibly counter automata. We prove that any visibly counter language satisfies the equations.

5.6.9 Definition

A visibly counter automaton (VCA) over a visibly pushdown alphabet A with threshold m is a tuple $(A, Q, q_0, F, \delta_0, \ldots, \delta_m)$ where

- A is a visibly pushdown alphabet
- Q is a finite set of states
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of final states
- $\delta_i \colon A \times Q \to Q$ for $i = 1, \ldots, m$ are the transition functions

We define the *stack-height* of a word $w \in A^*$, denoted by $\|w\|$ inductively:

- If $w \in A_C$ then $\|w\| = 1$,
- if $w \in A_R$ then $\|w\| = -1$,
- if $w \in A_I$ then $\|w\| = 0$, and
- if $w = w_1 \ldots w_n$ with $w_i \in A$, then $\|w\| = \sum_{i=1}^{n} \|w_i\|$.

Observe that a word w of length n is well-matched, if $\|w\| = 0$ and for each $i \leq n$, the condition $\|w_{<i}\| \geq 0$ holds.

A configuration of a VCA $\mathcal{A}$ is a tuple in $Q \times \mathbb{N}$ and we say that there exists an a-transition from (q, i) to (p, j), writing $(q, i) \xrightarrow{a} (p, j)$ if $j = i + \|a\|$ and $p = \delta_i(a, q)$. The run of a VCA $\mathcal{A}$ on a word $w \in A^{\triangle}$ of length n is the sequence of tuples

$$(q_1, i_1)(q_2, i_2) \ldots (q_n, i_n) \text{ with } (q_j, i_j) \xrightarrow{w_j} (q_{j+1}, i_{j+1}) \text{ and } q_1 = q_0.$$

A run is accepting, if $q_n \in F$ and the language accepted by $\mathcal{A}$ is the set of all words $w \in A^{\triangle}$ such that the run of $\mathcal{A}$ on w is accepting. Observe that we only consider well-matched words and it is hence not necessary to require $i_n = 0$ for an accepting run.

5.6.10 Proposition

If a language L is VCL, it satisfies equation 5.2.

Proof. Recall that equation 5.2 is satisfied by a language L if and only if for all $u, v, u', v' \in A^*$ such that uv, $u'v'$, $u'v$, $uv' \in A^{\triangle}$ the equality

$$\begin{aligned} \mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u',v'}(x)) &= \mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u,v'}(\mathrm{ext}^{\omega}_{u',v'}(x))) \\ &= \mathrm{ext}^{\omega}_{u,v}(\mathrm{ext}^{\omega}_{u',v}(\mathrm{ext}^{\omega}_{u',v'}(x))) \end{aligned}$$

holds in the syntactic Ext-algebra of L.

Assume that L is recognised by a VCA $\mathcal{A}$ with threshold m. For any $u \in A^*$ and $i \in \mathbb{N}$ such that $i + \|u\| \geq 0$, we define the function

$$r_{u,i} \colon Q \to Q \text{ where } r_{u,i}(q) = p \text{ iff } (q,i) \xrightarrow{u} (p, i + \|u\|).$$

Since $\mathcal{A}$ is a VCA with threshold m, for any two $i, j \geq m$, we have $r_{u,i} = r_{u,j}$. We show that if $\|u\| > 0$, then there exists an $s \in \mathbb{N}$ such that $r_{u^s,i} = r_{u^{2s},i}$ for all $i \in \mathbb{N}$. Observe that it suffices to prove that for each $i = 0, \ldots, m$ there exists an s_i with the property $r_{u^{s_i},i} = r_{u^{2s_i},i}$ and we obtain s as their least common multiple. Hence let $i \in \mathbb{N}$ be arbitrary but fixed and observe that

$$r_{u^2,i} = r_{u,i+\|u\|} \circ r_{u,i}$$

which implies that for $n \in \mathbb{N}$ such that $n \cdot \|u\| < m \leq (n+1) \cdot \|u\|$ and $l > 0$ the equality

$$r_{u^{n+l},i} = \underbrace{r_{u,m} \circ \cdots \circ r_{u,m}}_{l \text{ times}} \circ r_{u,i+n\|u\|} \circ \cdots \circ r_{u,i+\|u\|} \circ r_{u,i}$$

holds. Since the functions from Q to Q form a finite monoid, $r_{u,m}$ generates an idempotent such that for appropriate l we obtain

$$r_{u^{n+l},i} = r_{u^{n+2l},i}.$$

Letting $s = n \cdot l$ proves that claim. Similarly, if $\|u\| < 0$, then there also exists an $s \in \mathbb{N}$ such that $r_{u^s,i} = r_{u^{2s},i}$ for all $i \geq \left\|u^{2n}\right\|$ and also the case for $\|u\| = 0$ follows in the same fashion.

Let $u, v, u', v' \in A^*$ such that they can be inserted in the equation above and let s be their common exponent, in the sense that $r_{x^s,i} = r_{x^{2s},i}$ for $x \in \{u, u', v, v'\}$ and all $i \in \mathbb{N}$. This exponent exists, since we can again choose the least common multiple of all single exponents.

We observe that now for all $x \in A^{\triangle}$

$$r_{u^s x v^s,i} = r_{u^{2s} x v^{2s},i}$$

where $r_{u^s x v^s,0}(q_0)$ is a final state, if and only if the run of $\mathcal{A}$ on $u^s x v^s$ is accepting and hence

$$u^s x v^s \in L \Leftrightarrow u^{2s} x v^{2s} \in L$$

Since the exponent s was chosen independent of i we may even derive that the two words are syntactically equivalent with respect to L. Hence we obtain that the syntactic image of $u^s x v^s$ is $\mathrm{ext}^{\omega}_{u,v}(x)$.

Moreover, we derive that

$$r_{u^s u'^s x v'^s v^s,i} = r_{u^{2s} u'^s x v'^{2s} v^s,i} = r_{u^s u'^{2s} x v'^s v^{2s}},$$

which results in the syntactic equivalence of those words and by the previous observation in the validity of equation 5.2. ∎

It follows that neither the Ludwig language nor H^+ are VCLs. Moreover $\{a^n b^n c^m d^m \mid n, m \in \mathbb{N}\}$, satisfies the equation and it is not hard to construct a VCA recognising it.

5.7 Summary

This chapter was dedicated to finding a topological and algebraic theory for the visibly pushdown languages. We began by examining the structure of well-matched words and deriving of that structure a notion of recognition for languages of well-matched words: Ext-algebras. Subsequently, we derived the notion of morphisms between Ext-algebras and a notion of syntactic Ext-algebra. Together with these tools, it was possible to identify Ext-algebras as the adequate recognisers for visibly pushdown languages, that is, that a language is recognised by such an algebra if and only if it is visibly pushdown.

This realisation led to an Eilenberg-like theorem for the visibly pushdown languages: Classes of VPL with natural closure properties (pseudo-varieties) are in one-to-one correspondence with so called pseudo-varieties of Ext-algebras.

The topological perspective was then added, using a similar approach as that for the free profinite monoid, where it is defined via the completion of A^* as a metric space: We defined a metric on the set of well-matched words. In fact, it was possible to show that the completion of that space is the free profinite Ext-algebra in the sense that it also is equipped with operations of an Ext-algebra and as such also the Stone space of the visibly pushdown languages.

As a consequence, we were able to prove a Reiterman-like theorem for VPL, which states that each pseudo-variety of Ext-algebras is uniquely determined by a set of equations on the free profinite Ext-algebra.

Concluding this chapter, we gave a set of equations which is sound for a subclass of the visibly pushdown languages, namely the visibly counter languages and by adding additional equations, a set of equations that is sound for the languages that are an intersection of a regular language with the set of well-matched words $\mathcal{M}$Ext.

5.8 Further Research

While the results of this chapter, such as the construction of the free profinite Ext-algebra were proven very much from scratch, it should be considered whether they could not also be achieved using some of the heavier category-theoretical machinery, as for instance used in [UACM17] or the book of Almeida [Alm95].

That it is decidable whether, given a visibly pushdown automaton, its recognised language is equal to an intersection of a regular language with the set of well-matched words was shown by Löding et al. in [BLS06]. Finding a finite basis of equations that is complete for $\mathcal{M}$Ext would automatically imply decidability of the previous problem and thus present an algebraic reproof of the result of Löding et al., also improving the runtime of the decision algorithm.

An intermediate result that (to the best current knowledge of the author) seems very probable, is that the set of equations given for the visibly counter languages, is

already complete. If this conjecture should be true, it should be of help in reproving [BLS06].

A second direction of research would be the investigation of VPL in certain circuit-classes (or fragments of logic). This is motivated by the well-known characterisation of the regular languages in the class $\mathbf{AC}^0$ through the profinite equation $(x^{\omega-1}y)^{\omega+1} = (x^{\omega-1}y)^{\omega}$ for words x, y of equal length. Since the set of well-matched words is $\mathbf{TC}^0$-hard, this class seems to be a natural candidate for understanding the visibly pushdown languages contained in it. Understanding those languages, as in the case for the regular languages in $\mathbf{AC}^0$, may contribute to a better understanding of the class itself.

6

Typed Stamps and Projective Limits

To approach arbitrary Boolean algebras from a perspective of algebra and topology, we introduce the notion of typed stamps:

It is derived from typed monoids, which were originally introduced in [KLR05] to capture circuit classes and algebraic recognition of non-regular languages. The original definition included a monoid equipped with a finite Boolean algebra of subsets of the monoid. We replaced the finite Boolean algebra by a function from the monoid into a finite set. This definition is equivalent, saves some notation induced trouble and is more adapt from a topological perspective, which we develop and elaborate in later sections.

6.1 Typed Stamps

6.1.1 Definition

A typed stamp is a tuple $\mathsf{R} = (A, \mu, M, p, X)$, where A is a finite alphabet, M is a monoid, $\mu\colon A^* \to M$ is a monoid morphism, X is a finite set, and $p\colon M \to X$ is a function.

A language $L \subseteq A^*$ is recognised by a typed stamp $\mathsf{R} = (A, \mu, M, p, X)$ if there is a subset $C \subseteq X$, such that $L = (p \circ \mu)^{-1}(C)$. To abbreviate notation slightly, we let

$$\iota_{\mathsf{R}} = (p \circ \mu).$$

The set of languages recognised by R is denoted by $L(\mathsf{R})$. Observe that $L(\mathsf{R})$ always forms a finite Boolean algebra. Also, if R recognises a language L, then $L = \iota_{\mathsf{R}}^{-1}(\iota_{\mathsf{R}}(L))$.

A Reason for Typed Stamps

The reasons for endowing some infinite monoid with the additional structure of a morphism and a finite set of – figuratively speaking – accepting types, lie in complexity theory.

Fixing the morphism $\mu\colon A^* \to M$ is justified by a classical example, which was already mentioned in [PS05] as a reason to study stamps.

The language

$$L_{\text{even}} = \{w \in \{a,b\}^* \mid |w| \text{ is even.}\}$$

is recognised by the typed stamp

$$\mathsf{R} = (\{a,b\}^*, \tfrac{a\mapsto 1}{b\mapsto 1}, \mathbb{Z}_2, \mathrm{id}, \mathbb{Z}_2).$$

Any typed stamp recognising that language contains a copy of $\mathbb{Z}_2$ in the monoid component. But the monoid $\mathbb{Z}_2$ also recognises the language

$$L_{\text{parity}} = \{w \in \{a,b\}^* \mid w \text{ contains an even number of } a\text{'s.}\}$$

via the morphism that sends b to 0 and a to 1. It is a well-known result of [FSS84], that the languages recognised by $\mathbf{FO}[\mathcal{N}]$ (or equivalently $\mathbf{AC}^0$) do not contain L_{parity}, but do contain L_{even}, which justifies the necessity of fixing the morphism.

The finite set X and function $p\colon M \to X$ serve the purpose of restricting the expressive power of the monoid, by limiting the recognised languages. Why it is necessary in the non-regular case to enforce that restriction, if we aim at capturing natural classes of languages algebraically, is illustrated in the following example.

Let

$$L_{\text{Eq}} = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\},$$

which is recognised by the typed stamp

$$\mathsf{R} = (\{a,b\}, \tfrac{a\mapsto 1}{b\mapsto -1}, \mathbb{Z}, \chi_{\{0\}}, \{0,1\})$$

Then $L_{\text{Eq}} = \iota_{\mathsf{R}}^{-1}(\{1\})$. Note that the typed stamp R recognises finitely many languages: $\emptyset$, L_{Eq},L_{Eq}^c and $\{a,b\}^*$. Compare that to classical recognition: The monoid $\mathbb{Z}$ recognises any unary language L over $\{a\}^*$, since

$$L = h^{-1}\left(\{|w| \mid w \in L\}\right)$$

where h is the morphism sending a to 1. Since $\mathbb{Z}$ is the syntactic monoid of L_{Eq} and the morphism sending a to 1 and b to -1 is its syntactic morphism, any typed stamp recognising that language must, in the monoid component, contain a free monoid on one generator, that is generated by the image of a. Now, the class $\mathbf{Maj}[<]$ (a subclass of $\mathbf{TC}^0$) for instance, contains L_{Eq}, but it does not contain any unary encoding of an undecidable language. Hence the restriction of the accepting sets.

In their full generality, typed stamps can recognise very complex languages, such as

$$L_{\text{Prime}} = \{w \in \{a\}^* \mid |w| \text{ is prime.}\}.$$

which is recognised by the typed stamp

$$\mathsf{R} = (\{a\}, a \mapsto 1, \mathbb{N}, \chi_P, \{0,1\}),$$

where $P \subseteq \mathbb{N}$ is the set of all prime numbers. Then $L_{\text{Prime}} = \iota_{\mathsf{R}}^{-1}(\{1\})$.

Typed Stamps and Recognition

It is a well known and extremely useful characterisation, that a finite monoid M recognises some language L if and only if the syntactic monoid of L divides M. To establish a similar relationship between typed stamps and languages – or rather sets of languages – we introduce the notions of morphism and division of typed stamps.

Let $\mathsf{R} = (A, \mu, M, p, X)$ be a typed stamp. The *surjective restriction* of R is the typed stamp

$$\mathfrak{R}(\mathsf{R}) = (A, \mu, \mu(A^*), p, \iota_{\mathsf{R}}(A^*)).$$

If R is equal to $\mathfrak{R}(\mathsf{R})$, we say that R is restricted.

Remark: One could have required that in a typed stamp $\mathsf{R} = (A, \mu, M, p, X)$, both μ and p are surjective. For language recognition, this requirement makes no difference, but it is sometimes convenient to talk about a larger monoid without the morphism having to be surjective – for instance in the block product, which will be treated later. For proofs however, this is rather cumbersome, which is why we often assume the typed stamps to be restricted.

6.1.2 Definition

Let R and S be typed stamps with

$$\mathfrak{R}(\mathsf{R}) = (A, \mu, M, p, X) \text{ and } \mathfrak{R}(\mathsf{S}) = (B, \nu, N, q, Y).$$

A morphism of typed stamps $\Psi\colon \mathsf{R} \to \mathsf{S}$ consists of a triple

$$\Psi = (h, \phi, f),$$

where $h\colon A^* \to B^*$ and $\phi\colon M \to N$ are monoid morphisms and $f\colon X \to Y$ is a surjective function such that the diagram

$$\begin{array}{ccccc} A^* & \xrightarrow{\mu} & M & \xrightarrow{p} & X \\ {\scriptstyle h}\downarrow & & {\scriptstyle \phi}\downarrow & & {\scriptstyle f}\downarrow \\ B^* & \xrightarrow{\nu} & N & \xrightarrow{q} & Y \end{array}$$

commutes. By abuse of notation, we often omit mentioning the tuple explicitly and write, for instance, $\Psi\colon M \to N$ instead of $\phi\colon M \to N$.

For instance, the triple $\mathrm{id}_{\mathsf{R}} = (\mathrm{id}_{A^*}, \mathrm{id}_M, \mathrm{id}_X)$ is a morphism $\mathrm{id}_{\mathsf{R}}\colon \mathsf{R} \to \mathsf{R}$ – the identity morphism. Together with the composition of morphisms, which is the component wise composition, typed stamps contribute a category.

Remark: Observe that if morphisms were not defined on their surjective restrictions, then there would not exist a morphism from the typed stamp $(A, a \mapsto 1, \mathbb{Z}_2 \cup \{x\}, \mathrm{id}_{\mathbb{Z}_2}, \mathbb{Z}_2)$, where x is an absorbing element, to $(A, a \mapsto 1, \mathbb{Z}_2, \mathrm{id}_{\mathbb{Z}_2}, \mathbb{Z}_2)$, although they are almost the same and recognise the same languages.

6.1.3 Definition

Let R and S be typed stamps.

- S *factors through* R, if S and R are restricted and there exists a morphism of typed stamps $\Psi\colon \mathsf{R} \to \mathsf{S}$ with $\Psi = (\mathrm{id}_{A^*}, \phi, f)$, such that ϕ is a quotient morphism and f is surjective. We call Ψ the factoring morphism.
- S *divides* R, if $\mathfrak{R}(\mathsf{S})$ factors through $\mathfrak{R}(\mathsf{R})$. We write $\mathsf{S} \prec \mathsf{R}$.

6.1.4 Example

The typed stamp

$$\mathsf{S} = (\{a\}, a \mapsto 1, \mathbb{Z}_2, \chi_{\{1\}}, \{0, 1\}),$$

divides the typed stamp

$$\mathsf{R} = (\{a\}, a \mapsto 1, \mathbb{Z}, p, \{e, o\}),$$

where $p(x) = e$ if and only if x is even, since S factors through

$$\mathfrak{R}(\mathsf{R}) = (\{a\}, a \mapsto 1, \mathbb{N}, p, \{e, o\}),$$

where the quotient morphism $\phi\colon \mathbb{N} \to \mathbb{Z}_2$ sends even numbers to 0 and odd numbers to 1 and $f\colon \{e, o\} \to \{0, 1\}$ maps e to 0 and o to 1.

Restriction is to typed stamps, what submonoids are to monoids: When concerned with language recognition in finite monoids, we may always consider the submonoid induced by the recognising morphism. For typed stamps, we can assume that the typed stamp is restricted. Hence also division for typed stamps is similar to division of finite monoids: A typed stamp divides another, if it is the surjective morphic image of a sub.

6.1.5 Proposition

Let L be a language and R a typed stamp, then R recognises L if and only if $\mathfrak{R}(\mathsf{R})$ recognises L.

The proof is clear and thus omitted.

Unlike finite monoids, where two monoids recognise the same set of languages if and only if they are isomorphic, typed stamps come with some technical peculiarities: Two typed stamps might recognise the same languages even without being divisionally comparable.

6.1.6 Example

Let $\mathsf{R} = (A, \mu, M, p, X)$ be a typed stamp. Moreover, define the morphisms

$$\begin{aligned} \mu_2 \colon A^* &\to M \times \mathbb{Z}_2 \\ w &\mapsto (\mu(w), 1) \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mu_3 \colon A^* &\to M \times \mathbb{Z}_3 \\ w &\mapsto (\mu(w), 1) \end{aligned}$$

and let π_1 be the projection onto the first component. Then the typed stamps

$$(A^*, \mu_2, M \times \mathbb{Z}_2, (p \circ \pi_1), X) \text{ and } (A^*, \mu_3, M \times \mathbb{Z}_3, (p \circ \pi_1), X)$$

recognise the same languages as R, but do not divide each other.

We observe that in this example, the two typed stamps have unnecessarily big monoid components and the recognised languages are actually only dependent on M and X.

6.1.7 Definition

A typed stamp R is a *trivial extension* of another typed stamp S, if S divides R, where $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ is the factoring morphism $\Psi \colon \mathfrak{R}(\mathsf{R}) \to \mathfrak{R}(\mathsf{S})$ and f is a bijection. We write $\mathsf{S} \leq_T \mathsf{R}$.

For instance, the two typed stamps from the previous Example 6.1.6 are trivial extensions of R.

Also, the typed stamp R from Example 6.1.4 is a trivial extension of S. Both recognise the languages

$$L_{\text{even}} = \{w \in \{a\}^* \mid |w| \text{ is even.}\} \text{ and } L_{\text{odd}} = \{w \in \{a\}^* \mid |w| \text{ is odd.}\}.$$

Even though trivial extensions might posses extremely powerful monoid components they do not increase the expressive power of the typed stamp.

6.1.8 Proposition

Let R and S be typed stamps, then $\mathsf{S} \leq_T \mathsf{R}$ implies $L(\mathsf{R}) = L(\mathsf{S})$.

Proof. Without loss of generality, assume that R is restricted with $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (B, \nu, N, q, Y)$. Since $\mathsf{S} \leq_T \mathsf{R}$, there exist a morphism $\Psi \colon \mathsf{R} \to \mathsf{S}$ with $\Psi = (\mathrm{id}_{A^*}, \phi, f)$, such that f is a bijection. Let $C_Y \subseteq Y$, then

$$\iota_{\mathsf{S}}^{-1}(C_Y) = (f \circ \iota_{\mathsf{R}})^{-1}(C_Y),$$

which results in $L(\mathsf{S}) \subseteq L(\mathsf{R})$. Let $C_X \subseteq X$ and denote by $f^{-1} \colon Y \to X$ the inverse map of f. Since $(f \circ \iota_{\mathsf{R}}) = \iota_{\mathsf{S}}$ implies $\iota_{\mathsf{R}} = (f^{-1} \circ \iota_{\mathsf{S}})$, we obtain

$$\iota_{\mathsf{R}}^{-1}(C_X) = (f^{-1} \circ \iota_{\mathsf{S}})^{-1}(C_X)$$

and hence $L(\mathsf{S}) = L(\mathsf{R})$. ∎

Observe that Example 6.1.6 implies that the converse direction of the previous proposition is in general not true.

We introduce the notion of a syntactic typed stamp, with a slight alteration from the case known from finite monoids, since typed stamps are more adapt to recognising Boolean algebras, than to recognising single languages. For instance, the language $L = \{w \in A^* \mid |w| \equiv 1 \mod 4\}$ is recognised by the typed stamps

$$(A^*, \eta_L, \mathbb{Z}_4, \chi_{\{1\}}, \{0,1\}) \text{ and } (A^*, \eta_L, \mathbb{Z}_4, \mathrm{id}_{\mathbb{Z}_4}, \mathbb{Z}_4),$$

where η_L is the syntactic morphism of L. But the first recognises the Boolean algebra $\{\emptyset, L, L^c, A^*\}$ and the second the Boolean algebra generated by L and all its quotients by words.

6.1.9 Proposition

Let $\mathcal{B}$ be a finite Boolean algebra and let $u, v \in A^*$. The relation given by

$$u \sim_{\mathcal{B}} v \Leftrightarrow \forall L \in \mathcal{B}\ \forall x, y \in A^*\ (xuy \in L \Leftrightarrow xvy \in L)$$

is a congruence on A^*.

The proof is entirely unsurprising and thus omitted. We call

$$M_{\mathcal{B}} := A^* \backslash_{\sim_{\mathcal{B}}}$$

the syntactic monoid of $\mathcal{B}$ and the canonical quotient morphism $\eta_{\mathcal{B}} \colon A^* \to M_{\mathcal{B}}$ the syntactic morphism of $\mathcal{B}$.

Remark: Let R be a typed stamp with $\mathfrak{R}(\mathsf{R}) = (A, \mu, M, p, X)$. Then the atoms of $L(\mathsf{R})$ are precisely the languages of the form $\iota_{\mathsf{R}}^{-1}(x)$ for $x \in X$ and thus X is isomorphic to the Stone space of $L(\mathsf{R})$. We sometimes identify atoms from $L(\mathsf{R})$ and elements of X and write $x_L \in X$, meaning the unique element of X such that $L = \iota_{\mathsf{R}}^{-1}(x_L)$.

6.1.10 Definition

Let $\mathcal{B} \subseteq \mathcal{P}(A^*)$ be a finite Boolean algebra. Then the *syntactic typed stamp* of $\mathcal{B}$ is

$$\mathsf{R}(\mathcal{B}) = (A, \eta_{\mathcal{B}}, M_{\mathcal{B}}, p_{\mathcal{B}}, X_{\mathcal{B}})$$

where $M_{\mathcal{B}}$ is the syntactic monoid and $X_{\mathcal{B}}$ is the Stone space of $\mathcal{B}$. The map $\eta_{\mathcal{B}}$ is the syntactic morphism of $\mathcal{B}$ and the map $p_{\mathcal{B}}$ is given by

$$\forall\ x_L \in X_{\mathcal{B}}\ :\ p_{\mathcal{B}}(m) = x_L \Leftrightarrow m \in \eta_L(L)$$

It follows that

$$\iota_{\mathsf{R}(\mathcal{B})}(w) = x_L \Leftrightarrow w \in L.$$

The syntactic typed stamp of a Boolean algebra $\mathcal{B}$ is the divisionally smallest typed stamp recognising $\mathcal{B}$.

6.1.11 Proposition

A typed stamp R recognises a Boolean algebra $\mathcal{B}$ over A^* if and only if the syntactic stamp of $\mathcal{B}$ divides R.

Proof. Without loss of generality, assume that $\mathsf{R} = (A, \mu, M, p, X)$ is restricted. Let $\mathcal{B} \subseteq \mathcal{P}(A^*)$ such that R recognises $\mathcal{B}$. That is, there exists a Boolean algebra $\mathcal{B}_X$ of subsets of X, such that $\mathcal{B} = \{\iota_{\mathsf{R}}^{-1}(C) \mid C \in \mathcal{B}_X\}$. Conversely, since R recognises $\mathcal{B}$, $\mathcal{B}_X = \{\iota_{\mathsf{R}}(L) \mid L \in \mathcal{B}\}$.

We claim that there exists a quotient morphism $\phi\colon M \to M_{\mathcal{B}}$ and a function $f\colon X \to X_{\mathcal{B}}$ such that $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ is a factoring morphism $\Psi\colon \mathsf{R} \to \mathsf{R}(\mathcal{B})$.

We prove that $\eta_{\mathcal{B}}$ factors through μ. Let $u, v \in A^*$ such that $\mu(u) = \mu(v)$, which implies $\mu(xuy) = \mu(xvy)$ for all $x, y \in A^*$ and hence also $\iota_{\mathsf{R}}(xuy) = \iota_{\mathsf{R}}(xvy)$. We obtain, since R recognises $\mathcal{B}$, that for each $L \in \mathcal{B}$

$$xuy \in L \Leftrightarrow xvy \in L.$$

Hence the syntactic morphism $\eta_{\mathcal{B}}$ factors through μ and there exists a morphism $\phi\colon M \to M_{\mathcal{B}}$ such that $\eta_{\mathcal{B}} = \phi \circ \mu$.

Since $\mathcal{B}_X$ is isomorphic to $\mathcal{B}$ as a Boolean algebra $X_{\mathcal{B}}$ is isomorphic to $X_{\mathcal{B}_X}$ and there exists a function $f\colon X \to X_{\mathcal{B}}$, which is the dual of the inclusion $\mathcal{B}_X \subseteq \mathcal{P}(X)$. In particular, this implies that

$$\forall\ x_L \in X_{\mathcal{B}}\ :\ (f \circ \iota_{\mathsf{R}})(w) = x_L \Leftrightarrow w \in L,$$

which proves that $(f \circ \iota_{\mathsf{R}}) = \iota_{\mathsf{R}(\mathcal{B})}$.

Since μ and $\eta_{\mathcal{B}}$ are surjective, this implies that $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ is a factoring morphism, which proves one direction of the claim.

For the converse direction, assume that $\mathsf{R}(\mathcal{B})$ factors through R, that is there exists a morphism $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ with $\Psi\colon \mathsf{R} \to \mathsf{R}(\mathcal{B})$. Let $L \in \mathcal{B}$, then

$$L = \iota_{\mathsf{R}(\mathcal{B})}^{-1}(\iota_{\mathsf{R}(\mathcal{B})}(L)) = \iota_{\mathsf{R}}^{-1}(f^{-1}(\iota_{\mathsf{R}(\mathcal{B})}(L))),$$

which proves that R recognises L. ∎

If we want to characterise recognition of single languages rather than whole Boolean algebras, we define the syntactic typed stamp of a language $L \subseteq A^*$ to be the syntactic typed stamp of $\{\emptyset, L, L^c, A^*\}$, which is isomorphic to

$$\mathsf{R}(L) := (A^*, \eta_L, M_L, \chi_{\eta_L(L)}, \{0, 1\})$$

where η_L is the syntactic morphism and M_L the syntactic monoid of L.

We obtain as a Corollary:

6.1.12 Corollary

A typed stamp R recognises a language L if and only if the syntactic typed stamp of L divides R.

We define the product only for typed stamps over the same alphabet. This is sufficient for our purposes and preserves the property that the product recognises Boolean combinations of languages, as stated in the following proposition.

6.1.13 Definition Product of Typed Stamps

Let $\mathsf{R}_1 = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$ be two typed stamps. The product of R and S, is the typed stamp

$$\mathsf{R} \times \mathsf{S} := (A, \phi, M \times N, p \times q, X \times Y)$$

where $\phi\colon A^* \to M \times N$ is the morphism sending w to $(\mu(w), \nu(w))$.

6.1.14 Proposition

Let R and S be two typed stamps. The languages recognised by $\mathsf{R} \times \mathsf{S}$ are precisely the Boolean combinations of languages in $L(\mathsf{R})$ and $L(\mathsf{S})$.

Proof. Let $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$. Note that since X and Y are finite, any subset $C \subseteq X \times Y$ is of the form $\bigcup_{i=1}^{n} C_{X,i} \times C_{Y,i}$ where $C_{X,i} \subseteq X$ and $C_{Y,i} \subseteq Y$. It follows that, if $L = (\iota_{\mathsf{R}\times\mathsf{S}})^{-1}(C)$, then

$$\begin{aligned} L &= \bigcup_{i=1}^{n} (\iota_{\mathsf{R}\times\mathsf{S}})^{-1}(C_{X,i} \times C_{Y,i}) \\ &= \bigcup_{i=1}^{n} \iota_{\mathsf{R}}^{-1}(C_{X,i}) \cap \iota_{\mathsf{S}}^{-1}(C_{Y,i}). \end{aligned}$$

Hence every language recognised by the direct product $\mathsf{R} \times \mathsf{S}$ is a Boolean combination of languages recognised by R and S. To show that it recognises all Boolean combinations, suppose $L_X = \iota_{\mathsf{R}}^{-1}(C_X)$ and $L_Y = \iota_{\mathsf{S}}^{-1}(C_Y)$, then

$$L_X \cup L_Y = (\iota_{\mathsf{R}\times\mathsf{S}})^{-1}((C_X \times Y) \cup (X \times C_Y))$$

and

$$L_X \cap L_Y = (\iota_{\mathsf{R}\times\mathsf{S}})^{-1}(C_X \times C_Y).$$

That $\mathsf{R} \times \mathsf{S}$ recognises complements of languages in $L(\mathsf{R})$ and $L(\mathsf{S})$ follows directly from the previous observation and the fact that both $L(\mathsf{R})$ and $L(\mathsf{S})$ are Boolean algebras. ∎

6.1.15 Corollary

Let $\mathsf{R}_1, \mathsf{R}_2$ and $\mathsf{S}_1, \mathsf{S}_2$ be typed stamps, then $L(\mathsf{R}_1) = L(\mathsf{R}_2)$ and $L(\mathsf{S}_1) = L(\mathsf{S}_2)$ implies $L(\mathsf{R}_1 \times \mathsf{R}_2) = L(\mathsf{S}_1 \times \mathsf{S}_2)$.

We can now give an exact dependence between trivial extensions and recognition of languages.

6.1.16 Proposition

Let R and S be typed stamps, then $L(\mathsf{R}) = L(\mathsf{S})$ if and only if there exists a typed stamp T such that $\mathsf{T} \leq_T \mathsf{S}$ and $\mathsf{T} \leq_T \mathsf{R}$.

Proof. If there exists a typed stamp with $\mathsf{T} \leq_T \mathsf{S}$ and $\mathsf{T} \leq_T \mathsf{R}$, then $L(\mathsf{S}) = L(\mathsf{R})$ follows immediately from Proposition 6.1.8.

For the converse direction, let $\mathcal{B} = L(\mathsf{R}) = L(\mathsf{S})$.

We prove that R is a trivial extension of the syntactic typed stamp $\mathsf{R}(\mathcal{B})$. Without loss of generality, assume that $\mathsf{R} = (A, \mu, M, p, X)$ is restricted. By Proposition 6.1.11, $\mathsf{R}(\mathcal{B})$ divides R and since R is restricted, there exists a factoring morphism $\Psi\colon \mathsf{R} \to \mathsf{R}_{\mathcal{B}}$ with $\Psi = (\mathrm{id}_{A^*}, \phi, f)$.

Since both typed stamps recognise precisely the same languages, f must be a bijection. Hence $\mathsf{R}(\mathcal{B}) \leq_T \mathsf{R}$.

Replacing R by S gives that $\mathsf{R}(\mathcal{B}) \leq_T \mathsf{S}$, which proves the claim. ∎

6.2 Streams of Typed Stamps

As illustrated before, whether a language is recognised by a formula in $\mathbf{FO}[\mathcal{N}]$ depends rather on the syntactic morphism of a language than on its syntactic monoid. It is thus, that we study classes of typed stamps whose monoid morphisms belong to certain classes of morphisms. When regarding classes of languages, it becomes important to be able to switch alphabets, which is mostly done via (inverse) morphisms between free monoids. However, in some cases, we may not regard all morphisms between free monoids, since for instance

$$\{w \in \{a,b\}^* \mid |w|_a \equiv 0 \mod 2\} = \phi^{-1}(\{w \in \{0,1\}^* \mid |w| \equiv 0 \mod 2\})$$

where $\phi\colon \{a,b\}^* \to \{0,1\}^*$ is the morphism sending b to λ and a to 1. The language consisting of all words with an even number of as is not contained in $L(\mathbf{FO}[\mathcal{N}])$, but the language consisting of all words of even length is. With that in mind, we consider classes of morphisms $\mathcal{C}$ between free monoids having the following properties:

1. $\mathcal{C}$ is closed under composition: If A, B and C are finite alphabets and $h\colon A^* \to B^*$ and $g\colon B^* \to C^*$ are morphisms in $\mathcal{C}$, then $(g \circ h) \in \mathcal{C}$.
2. $\mathcal{C}$ contains all length-preserving morphisms, where a morphism $h\colon A^* \to B^*$ is said to be length-preserving, if $|h(a)| = 1$ for each $a \in A$.

In particular, the identity is always contained in $\mathcal{C}$. While the necessity of the second requirement is not immediately clear, it is already justified in [PS05] for smoothing out some difficulties – for instance, it ensures the identity is always contained – and

does not pose a significant restriction, as up to now, all known interesting classes are closed under inverse length-preserving morphisms.

The class of all morphisms between free monoids is denoted by *all* and the class of all length-preserving morphisms by *lp*.

Further instances of classes of morphisms that satisfy the requirements above are

- the class of non-erasing morphisms, short *ne*: A morphism $h\colon A^* \to B^*$ is called non-erasing, if $h(a) \neq \lambda$ for all $a \in A$.
- the length-multiplying morphisms, short *lm*: A morphism $h\colon A^* \to B^*$ is called length-multiplying, if $|h(a)| = k$ for all $a \in A$.

Any morphism of free monoids induces a morphism of typed stamps in the following way:

6.2.1 Definition

Let $\varphi\colon A^* \to B^*$ be a morphism in $\mathcal{C}$ and let $\mathsf{S} = (B, \nu, N, q, Y)$ be a typed stamp. Then, we denote by $\mathsf{S} \cdot \varphi$ the typed stamp

$$(A, \nu \circ \varphi, N, q, Y).$$

We say that $\mathsf{S} \cdot \varphi$ is a $\mathcal{C}$-transformation of S.

In particular, φ induces a morphism of typed stamps $\Psi_\varphi\colon \mathsf{S} \cdot \varphi \to \mathsf{S}$ with $\Psi_\varphi = (\varphi, \mathrm{id}_N, \mathrm{id}_Y)$.

Let $\mathbf{V}$ be a class of typed stamps and A a finite alphabet. Then $\mathbf{V}(A)$ is the class of all typed stamps $\mathsf{R} \in \mathbf{V}$ over the alphabet A, that is the class of typed stamps of the form $\mathsf{R} = (A, \mu, M, p, X)$ in $\mathbf{V}$.

6.2.2 Definition Stream of Typed Stamps

A class of typed stamps $\mathbf{V}$ is a $\mathcal{C}$-stream of typed stamps, if $\mathbf{V}$ is closed under

1. division: if $\mathsf{R} \in \mathbf{V}$ and $\mathsf{S} \prec \mathsf{R}$, then $\mathsf{S} \in \mathbf{V}$,
2. finite direct products: if $\mathsf{R}, \mathsf{S} \in \mathbf{V}$, then $\mathsf{R} \times \mathsf{S} \in \mathbf{V}$,
3. $\mathcal{C}$-transformations: if $\mathsf{R} \in \mathbf{V}(A)$ and $\varphi\colon B^* \to A^*$ is a $\mathcal{C}$-morphism, then $\mathsf{R} \cdot \varphi \in \mathbf{V}$
4. and trivial extensions: if $\mathsf{R} \in \mathbf{V}$ and $\mathsf{R} \leq_T \mathsf{S}$, then $\mathsf{S} \in \mathbf{V}$.

Observe that any set of typed stamps generates a $\mathcal{C}$-stream of typed stamps, by closing under the operations above.

6.2.3 Example

One may verify that the typed stamps $(A, \mu, M, \mathrm{id}_M, M)$, where A is some arbitrary alphabet and M is a finite monoid, generate an *all*-stream of typed stamps.
This *all*-stream consists of the typed stamps (A, μ, M, p, X) where A is some alphabet, M a monoid, X is a finite monoid and p is a monoid morphism.

This stream in particular has additional structure. Observe that the class of languages recognised by the element of that *all*-stream are precisely the regular languages, which are closed under quotients by words. This structure is mirrored on the side of typed stamps as follows:

For a monoid M, a set X, some element $m \in M$ and a function $p\colon M \to X$, we define the functions $(m \cdot p)\colon M \to X$ and $(p \cdot m)\colon M \to X$ by letting

$$(m \cdot p)(x) = p(m \cdot x) \text{ and } (p \cdot m)(x) = p(x \cdot m).$$

6.2.4 Definition

Let $\mathsf{R} = (A, \mu, M, p, X)$ be a typed stamp, then we define the typed stamps

$$a^{-1}\mathsf{R} := (A, \mu, M, \mu(a) \cdot p, X),$$

and

$$\mathsf{R}a^{-1} := (A, \mu, M, p \cdot \mu(a), X).$$

6.2.5 Proposition

Let R be a typed stamp recognising a language L, then $a^{-1}L$ (resp. La^{-1}) is recognised by the typed stamp $a^{-1}\mathsf{R}$ (resp. $\mathsf{R}a^{-1}$).

Proof. Suppose that L is recognised by $\mathsf{R} = (A, \mu, M, p, X)$. Then, $L = \iota_{\mathsf{R}}^{-1}(C)$ for some $C \subseteq X$. Recall that, for $a \in A$ and a subset $N \subseteq M$, we defined

$$\mu(a)^{-1}N = \{m \in M \mid \mu(a)m \in N\}$$

and thus

$$\begin{aligned} a^{-1}L &= \{w \in A^* \mid aw \in L\} \\ &= \{w \in A^* \mid aw \in \iota_{\mathsf{R}}^{-1}(C)\} \\ &= \{w \in A^* \mid \mu(a)\mu(w) \in p^{-1}(C)\} \\ &= \{w \in A^* \mid \mu(w) \in \mu(a)^{-1}p^{-1}(C)\} \\ &= \mu^{-1}(\mu(a)^{-1}p^{-1}(C)) \end{aligned}$$

In a similar fashion, it follows that

$$\mu(a)^{-1}p^{-1}(C) = (\mu(a) \cdot p)^{-1}(C).$$

Hence $a^{-1}L = ((\mu(a)\cdot p)\circ\mu)^{-1}(C) = \iota_{a^{-1}\mathsf{R}}^{-1}(C)$. The case for La^{-1} follows accordingly and hence the claim holds. ∎

It follows from Proposition 6.2.5, that $L(a^{-1}\mathsf{R}) = \{a^{-1}L \mid L \in L(\mathsf{R})\}$.

6.2.6 Definition Pseudo-Variety of Typed Stamps

A $\mathcal{C}$-stream of typed stamps $\mathbf{V}$ is called a $\mathcal{C}$-pseudo-variety of typed stamps, if for each finite alphabet A, each $\mathsf{R} \in \mathbf{V}(A)$ and each $a \in A$, we have

$$a^{-1}\mathsf{R} \in \mathbf{V}(A) \text{ and } \mathsf{R}a^{-1} \in \mathbf{V}(A).$$

For instance, the *all*-stream from Example 6.2.3 is also an *all*-pseudo-variety.

6.3 Eilenberg for Streams of Typed Stamps

The proofs in this section are often adaptions of the proofs in [Str02] (Eilenberg for stamps) or [Pin16] (Eilenberg for finite monoids), which were modified to work for typed stamps.

We prove that each $\mathcal{C}$-stream of typed stamps $\mathbf{V}$ corresponds to a class of languages $\mathcal{V}$ with certain closure properties.

6.3.1 Definition $\mathcal{C}$-Stream of Languages

A $\mathcal{C}$-stream of languages $\mathcal{V}$ is a mapping such that for each finite alphabet A,

1. $\mathcal{V}(A)$ is a Boolean algebra of languages over A^* and
2. $\mathcal{V}$ is closed under inverse $\mathcal{C}$-morphisms, that is if B is a finite alphabet and $\varphi\colon A^* \to B^*$ is a $\mathcal{C}$-morphism, then $L \in \mathcal{V}(B)$ implies $\varphi^{-1}(L) \in \mathcal{V}(A)$.

We define the correspondence $\mathbf{V} \to \mathcal{V}$ as follows: Given a $\mathcal{C}$-stream of typed stamps $\mathbf{V}$, we let $\mathcal{V}$ be the class of languages recognised by members of $\mathbf{V}$ and write $\mathbf{V} \mapsto \mathcal{V}$.

6.3.2 Proposition

If $\mathbf{V} \mapsto \mathcal{V}$, then $\mathcal{V}$ is a $\mathcal{C}$-stream of languages.

Proof. Let A be a finite alphabet. Since $\mathbf{V}$ is closed under finite direct products, it follows from Proposition 6.1.14 and the definition of $\mathcal{V}$ as the class of all elements recognised by members of $\mathbf{V}$, that $\mathcal{V}(A)$ is a Boolean algebra.

To show that if $\varphi\colon A^* \to B^*$ is a $\mathcal{C}$-morphism and $L \in \mathcal{V}(B)$, then $\varphi^{-1}(L) \in \mathcal{V}(A)$, it suffices to see that if L is recognised by S then $\mathsf{S}\cdot\varphi$ recognises $\varphi^{-1}(L)$. ∎

6.3.3 Proposition

The correspondence $\mathbf{V} \to \mathcal{V}$ is one-to-one: If $\mathbf{V} \mapsto \mathcal{V}$ and $\mathbf{W} \mapsto \mathcal{W}$, then $\mathbf{V} = \mathbf{W}$ if and only if $\mathcal{V} = \mathcal{W}$.

Proof. Assume that $\mathbf{V} = \mathbf{W}$, then it follows by definition that $\mathcal{V} = \mathcal{W}$.

For the converse direction, assume that $\mathcal{V} \subseteq \mathcal{W}$. We show that this implies $\mathbf{V} \subseteq \mathbf{W}$.

Let $\mathsf{R} \in \mathbf{V}$ with $\mathsf{R} = (A, \mu, M, p, X)$ and let $X = \{x_1, \ldots, x_k\}$ for $k \in \mathbb{N}$. Moreover, let $L_i = \iota_{\mathsf{R}}^{-1}(\{x_i\})$ for $i = 1, \ldots, k$. It follows from Proposition 6.1.14, that

$$L(\mathsf{R}) = L\left(\mathsf{R}(L_1) \times \cdots \times \mathsf{R}(L_k)\right).$$

By Proposition 6.1.16, this implies that there exists a typed stamp T such that both the product and R are trivial extensions of T.

Observe that since $L_i \in \mathcal{V}(A)$, we have $L_i \in \mathcal{W}(A)$ and hence there exists a typed stamp in $\mathbf{W}$ which recognises L_i. Since $\mathbf{W}$ is closed under division, by Proposition 6.1.11, we obtain $\mathsf{R}(L_i) \in \mathbf{W}$ and since $\mathbf{W}$ is closed under finite direct products, $\mathsf{R}(L_1) \times \cdots \times \mathsf{R}(L_k) \in \mathbf{W}$.

Then T is in $\mathbf{W}$ since it divides the product and $\mathbf{W}$ is closed under division and hence R is also in $\mathbf{W}$, since it is a trivial extension of T and $\mathbf{W}$ is closed under trivial extension.

The equality follows by symmetry of the argument. ∎

The correspondence $\mathcal{V} \to \mathbf{V}$ is defined in the following way: Given a $\mathcal{C}$-stream of languages $\mathcal{V}$, we let $\mathbf{V}$ be the stream of typed stamps generated by the syntactic typed stamps of languages in $\mathcal{V}$, that is, the smallest class of typed stamps, that contains all syntactic typed stamps and is closed under division, finite direct products, $\mathcal{C}$-transformations and trivial extension. We write $\mathcal{V} \mapsto \mathbf{V}$.

6.3.4 Theorem

The correspondences $\mathbf{V} \to \mathcal{V}$ and $\mathcal{V} \to \mathbf{V}$ are mutually inverse bijective correspondences between $\mathcal{C}$-streams of typed stamps and $\mathcal{C}$-streams of languages.

Proof. We first prove that $\mathcal{V} \mapsto \mathbf{V}$ and $\mathbf{V} \mapsto \mathcal{W}$ implies $\mathcal{V} = \mathcal{W}$.

Let $\mathcal{V}$ be a $\mathcal{C}$-stream of languages such that $\mathcal{V} \mapsto \mathbf{V}$ and $\mathbf{V} \mapsto \mathcal{W}$. If $L \in \mathcal{V}(A)$, then the syntactic typed stamp of L is contained in $\mathbf{V}$ and since $\mathcal{W}(A)$ consists of all languages recognised by some typed stamp in $\mathbf{V}(A)$, $L \in \mathcal{W}(A)$, which proves $\mathcal{V}(A) \subseteq \mathcal{W}(A)$ for each finite alphabet A.

To prove the inclusion $\mathcal{W}(A) \subseteq \mathcal{V}(A)$, assume that $L \in \mathcal{W}(A)$. Then, by definition of $\mathcal{W}(A)$, L is recognised by some typed stamp in $\mathbf{V}(A)$.

Before we can proceed, we need the following Lemma.

6.3.5 Lemma

Let $L \subseteq A^*$ and let R, R_T, S, S_T T, and T_T be typed stamps over A such that $\mathsf{R} \leq_T \mathsf{R}_T$, $\mathsf{S} \leq_T \mathsf{S}_T$ and $\mathsf{T} \leq_T \mathsf{T}_T$. Moreover let B be an alphabet and $\varphi\colon B^* \to A^*$ be a morphism. Then the implications

- $R(L) \prec \mathsf{R}_T \Rightarrow R(L) \prec \mathsf{R}$,
- $R(L) \prec \mathsf{R}_T \times \mathsf{S}_T \Rightarrow R(L) \prec \mathsf{R} \times \mathsf{S}$, and
- $R(L) \prec \mathsf{T}_T \cdot \varphi \Rightarrow R(L) \prec \mathsf{T} \cdot \varphi$.

hold.

Proof. Follows immediately from Proposition 6.1.15 and 6.1.8 and the observation that $L(\mathsf{T}_T \cdot \varphi) = L(\mathsf{T} \cdot \varphi)$. ∎

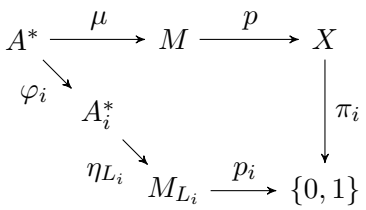
Now by definition of $\mathbf{V}$ as the smallest $\mathcal{C}$-stream generated by the syntactic monoids of languages in $\mathcal{V}$ and Lemma 6.3.5, there exists an $n \in \mathbb{N}$, finite alphabets A_i, $\mathcal{C}$-morphisms $\varphi_i\colon A^* \to A_i^*$ and languages $L_i \in \mathcal{V}(A_i)$ for $i \in \{1, \ldots, n\}$ such that $\mathsf{R}(L)$ divides the direct product

$$\mathsf{R}(L_1) \cdot \varphi_1 \times \cdots \times \mathsf{R}(L_n) \cdot \varphi_n \tag{6.1}$$

Note that it is due to Lemma 6.3.5, that we do not need to consider trivial extensions. We let $\mathsf{R} = (A^*, \mu, M, p, X)$ be the restriction of the typed stamp in (6.1). Then, since $\mathsf{R}(L)$ divides R, there exists a set $C \subseteq X$ such that $L = \iota_{\mathsf{R}}^{-1}(C)$ and since C is finite

$$L = \bigcup_{x \in C} \iota_{\mathsf{R}}^{-1}(x)$$

To show that $L \in \mathcal{V}(A)$, it thus suffices to show that for each $x \in X$, it holds that $\iota_{\mathsf{R}}^{-1}(\{x\}) \in \mathcal{V}(A)$. Denote by $\pi_i\colon X \to \{0,1\}$ the ith projection sending $(x_1, \ldots, x_n)$ to x_i – recall that $x_i \in \{0,1\}$. Then the diagram



commutes.

Since each $x \in X$ is of the form $x = (x_1, \ldots, x_n)$, where $x_i \in \{0,1\}$ for $i \in \{1, \ldots, n\}$, we get $\{x\} = \bigcap_{i=1}^{n} \pi_i^{-1}(x_i)$ and hence

$$\iota_{\mathsf{R}}^{-1}(x) = \bigcap_{i=1}^{n} (\pi_i \circ \iota_{\mathsf{R}})^{-1}(x_i) = \bigcap_{i=1}^{n} (\iota_{\mathsf{R}(L_i)} \circ \varphi_i)^{-1}(x_i)$$

Now $\mathcal{V}(A)$ is a $\mathcal{C}$-stream of languages and hence it suffices to show that $\iota^{-1}_{\mathsf{R}(L_i)}(x_i)$ is in $\mathcal{V}(A_i)$. But since $\iota^{-1}_{\mathsf{R}(L_i)}(x_i)$ is either L_i or L_i^c depending on whether $x_i = 0$ or $x_i = 1$ that claim holds and it follows that $L \in \mathcal{V}(A)$, which proves $\mathcal{V} = \mathcal{W}$.

To see that $\mathbf{V} \mapsto \mathcal{V}$ and $\mathcal{V} \mapsto \mathbf{W}$, implies $\mathbf{V} = \mathbf{W}$, observe that if

$$\mathbf{V} \mapsto \mathcal{V} \mapsto \mathbf{W} \mapsto \mathcal{W}$$

then the previous observation shows $\mathcal{V} = \mathcal{W}$ and by Proposition 6.3.3, $\mathbf{V} = \mathbf{W}$. ■

If a stream of typed stamps $\mathbf{V}$ corresponds to a stream of languages $\mathcal{V}$, we write $\mathbf{V} \leftrightarrow \mathcal{V}$.

6.3.6 Proposition

Let $\mathbf{V} \leftrightarrow \mathcal{V}$, then $\mathcal{V}$ is a $\mathcal{C}$-stream closed under quotients if and only if $\mathbf{V}$ is a $\mathcal{C}$-pseudo-variety.

Proof. Suppose that $L \in \mathcal{V}$ and $\mathbf{V}$ is a $\mathcal{C}$-pseudo-variety. Then, there exists a typed stamp $\mathsf{R} \in \mathbf{V}$ recognising L. It follows from Proposition 6.2.5 that $a^{-1}\mathsf{R}$ recognises $a^{-1}L$ and since $\mathbf{V}$ is a $\mathcal{C}$-pseudo-variety, $a^{-1}\mathsf{R} \in \mathbf{V}$.

For the converse direction, suppose $\mathcal{V}$ is a $\mathcal{C}$-stream closed under quotients. We first show that $\mathsf{R}(L) \in \mathbf{V}$ implies $a^{-1}\mathsf{R}(L) \in \mathbf{V}$.

Suppose that $\mathsf{R}(L) \in \mathbf{V}$, then $L \in \mathcal{V}$ and since $\mathcal{V}$ is closed under quotients, $\mathsf{R}(a^{-1}L)$ is in $\mathbf{V}$. It is not hard to see, that $\mathsf{R}(a^{-1}L)$ and $a^{-1}\mathsf{R}(L)$ recognise the same set of languages, that is the Boolean algebra generated by $a^{-1}L$.

Proposition 6.1.16 implies that there exists a typed stamp T, such that $\mathsf{T} \leq_T a^{-1}\mathsf{R}(L)$ and $\mathsf{T} \leq_T \mathsf{R}(a^{-1}L)$. Since $\mathsf{R}(a^{-1}L) \in \mathbf{V}$, T is contained in $\mathbf{V}$ by closure under quotients and $a^{-1}\mathsf{R}(L)$ is thus contained in $\mathbf{V}$ by closure under trivial extension.

Now let $\mathsf{R} \in \mathbf{V}$ be a typed stamp. We claim that $a^{-1}\mathsf{R} \in \mathbf{V}$. Observe that if $x \in X$, then $\iota^{-1}_{\mathsf{R}}(x) = L$ implies $\iota^{-1}_{a^{-1}\mathsf{R}}(x) = a^{-1}L$. Since $L \mapsto a^{-1}L$ distributes over Boolean operations, we obtain

$$L(a^{-1}\mathsf{R}) = \{a^{-1}L \mid L \in L(\mathsf{R})\}$$

Let $X = \{x_1, \ldots, x_k\}$ and $L_i = \iota^{-1}_{\mathsf{R}}(x_i)$ for $i \in \{1, \ldots, n\}$. Then

$$L(\mathsf{R}) = L(\mathsf{R}(L_1) \times \cdots \times \mathsf{R}(L_k))$$

by Proposition 6.1.14 and

$$L(a^{-1}\mathsf{R}) = L(a^{-1}\mathsf{R}(L_1) \times \cdots \times a^{-1}\mathsf{R}(L_k)).$$

Since $\mathsf{R}(L_i) \in \mathbf{V}$, we obtain $a^{-1}\mathsf{R}(L_i) \in \mathbf{V}$ for each $i \in \{1, \ldots, k\}$ and thus their direct product is also in $\mathbf{V}$. Again by 6.1.16 and closure under quotients and trivial extensions, $a^{-1}\mathsf{R}$ is thus in $\mathbf{V}$, which proves the claim. ■

Inherent Classes of Typed Stamps

Consider the following example:

6.3.7 Example

We say that a language $L \subseteq A^*$ is symmetrical, if for any word $w \in L$, any permutation of the letters in w also contributes a word in L. For instance, if $abba \in L$, then also $abab, bbaa, \ldots \in L$. For a finite alphabet A, denote by $\mathcal{V}(A)$ the set of all symmetrical languages. Then the class $\mathcal{V}$ is an *all*-stream, which also contains non-regular languages, such as

$$L_{\mathrm{Eq}} = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}.$$

The corresponding *all*-stream of typed stamps contains, for instance, the typed stamp

$$\mathsf{R} = (\{a,b\}^*, \mathrm{id}, \{a,b\}^*, \chi_{L_{\mathrm{Eq}}}, \{0,1\})$$

The syntactic monoids of the languages in $\mathcal{V}(A)$, however, all are commutative.

By trivial extension, a lot of unnecessarily large monoids are introduced in the $\mathcal{C}$-stream. This is unavoidable, if we want to keep the property, that any set of typed stamps generates a $\mathcal{C}$-stream. Also, being able to consider a monoid, that is just a little too large – for instance in the direct product – is sometimes rather convenient. It is however evident, that a lot of these large monoid components contribute in general little to the algebraic understanding of the languages. We will thus consider typed stamps that are minimal with respect to trivial extension in the following sense:

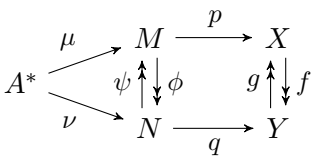
6.3.8 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of languages. Then division and trivial extension are partial orders on $\mathbf{V}$.

Proof. Let $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$ be typed stamps. We prove that division is a partial order on $\mathbf{V}$. Recall that $\mathsf{S} \prec \mathsf{R}$ if and only if $\mathfrak{R}(\mathsf{S})$ factors through $\mathfrak{R}(\mathsf{R})$. We assume without loss of generality, that R and S are restricted.

It follows immediately that $\prec$ is reflexive, since R factors through R with the identity being the factoring morphism.

To show that it is antisymmetric, assume that $\mathsf{S} \prec \mathsf{R}$ and $\mathsf{R} \prec \mathsf{S}$ that is, there exist morphisms $\Psi\colon \mathsf{S} \to \mathsf{R}$ and $\Phi\colon \mathsf{R} \to \mathsf{S}$, with $\Psi = (\mathrm{id}_{A^*}, \psi, g)$ and $\Phi = (\mathrm{id}_{A^*}, \phi, f)$. Summarising the situation in a commutative diagram, we have:



We prove that $\Psi \circ \Phi = \mathrm{id}_{\mathsf{S}}$ and $\Psi \circ \Phi = \mathrm{id}_{\mathsf{R}}$, by proving that the components are mutually inverse bijections.

Let $m \in M$ and $n \in N$. From the commutativity of the diagram, we obtain

$$m = (\mu \circ \nu^{-1} \circ \phi)(m) = (\psi \circ \phi)(m)$$

and

$$n = (\nu \circ \mu^{-1} \circ \psi)(n) = (\phi \circ \psi)(n).$$

Thus ϕ and ψ are inverse to each other.

Similarly, f and g are inverse to each other and we conclude that since M and N and X and Y are isomorphic, that (modulo isomorphism) $\mu = \nu$ and $f = g$ and thus $\mathsf{S} = \mathsf{R}$.

Transitivity follows immediately from composing the factoring morphisms.

Since $\mathsf{R} \leq_T \mathsf{S}$ requires that $\mathsf{R} \prec \mathsf{S}$ and all arguments above still hold, if f and g are required to be bijections, it follows that trivial extension defines a partial order on $\mathbf{V}$. ∎

Observe that the minimal elements of a $\mathcal{C}$-stream $\mathbf{V}$ with respect to division are not very interesting, since the trivial typed stamp

$$\mathsf{R} = (A^*, (a \mapsto 1)_{a \in A}, 1_M, \chi_{\{1\}}, \{1\}),$$

where 1_M is the trivial monoid, divides every typed stamp.

There is an exact characterisation of the typed stamps in $\mathbf{V}$ minimal with respect to trivial extension.

6.3.9 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps and $\mathsf{R} \in \mathbf{V}$. Then R is minimal with respect to trivial extension if and only if R is the syntactic typed stamp of $L(\mathsf{R})$.

Proof. Let $\mathcal{B} = L(\mathsf{R})$.

Assume that R is minimal with respect to trivial extension, then $\mathsf{R}(\mathcal{B})$ divides R and since they recognise the same set of languages $\mathsf{R}(\mathcal{B}) \leq_T \mathsf{R}$ which implies that they are equal, since R is minimal.

For the other direction, assume that R is the syntactic typed stamp of $\mathcal{B}$ and that there exists a typed stamp S with $\mathsf{S} \leq_T \mathsf{R}$. Then $L(\mathsf{S}) = \mathcal{B}$ and by Proposition 6.1.11 also $\mathsf{R} \leq_T \mathsf{S}$, which implies $\mathsf{R} = \mathsf{S}$, hence R is minimal. ∎

6.3.10 Definition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps and define

$$\mathcal{I}(\mathbf{V}) = \{\mathsf{R} \in \mathbf{V} \mid \mathsf{R} \text{ is minimal with respect to trivial extension.}\}.$$

and $\mathcal{I}_A(\mathbf{V}) := \mathcal{I}(\mathbf{V}) \cap \mathbf{V}(A)$. We call $\mathcal{I}(\mathbf{V})$ the *inherent* class of $\mathbf{V}$.

As a direct consequence of the previous proposition, we can characterise $\mathcal{I}_A(\mathbf{V})$ in terms of languages as follows:

6.3.11 Corollary

Let A be a finite alphabet and $\mathbf{V}$ a $\mathcal{C}$-stream of typed stamps with $\mathbf{V} \leftrightarrow \mathcal{V}$, then

$$\mathcal{I}_A(\mathbf{V}) = \{\mathsf{R}(\mathcal{B}) \mid \mathcal{B} \text{ is a finite Boolean subalgebra of } \mathcal{V}(A).\}$$

We can now state as a consequence of the Eilenberg theorem for $\mathcal{C}$-streams of typed stamps an Eilenberg theorem for inherent classes of typed stamps.

6.3.12 Corollary

There is a one-to-one correspondence between $\mathcal{C}$-streams of languages and inherent classes of $\mathcal{C}$-streams of typed stamps.

Proof. Let $\mathbf{V}$ and $\mathbf{W}$ be $\mathcal{C}$-streams of typed stamps. We prove that

$$\mathbf{V} = \mathbf{W} \Leftrightarrow \mathcal{I}(\mathbf{V}) = \mathcal{I}(\mathbf{W}).$$

The direction from left to right is obvious. For the converse direction, observe that by Corollary 6.3.11 the class of languages recognised by $\mathcal{I}(\mathbf{V})$ (resp. $\mathcal{I}(\mathbf{W})$) is precisely the stream of languages corresponding to $\mathbf{V}$ (resp. $\mathbf{W}$) and hence $\mathbf{V} = \mathbf{W}$ by Theorem 6.3.4. ∎

6.4 Projective Limits of Streams

To achieve a Reiterman-like characterisation of streams of typed stamps, we make use of projective limits. Since trivial extensions have some undesirable properties – like insufficient algebraic information about the languages – that the projective limit of a $\mathcal{C}$-stream would inevitably inherit, we are going to define it via inherent varieties.

Projective Systems

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of stamps. Since division is a partial order on $\mathbf{V}$, the inherent $\mathcal{C}$-stream $\mathcal{I}(\mathbf{V})$ inherits this order, which makes it a poset.

We make frequent use of a property, which is not particularly hard to verify: If $\mathsf{S}, \mathsf{R} \in \mathcal{I}_A(\mathbf{V})$, then $\mathsf{S} \prec \mathsf{R}$ if and only if S factors through R, in particular the factoring morphism $\Psi\colon \mathsf{R} \to \mathsf{S}$ is uniquely determined.

6.4.1 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps, then $\mathcal{I}_A(\mathbf{V})$ is a directed poset.

Proof. Let $\mathsf{R}_1, \mathsf{R}_2 \in \mathcal{I}_A(\mathbf{V})$ and let $\mathcal{V}$ be the $\mathcal{C}$-stream of languages corresponding to $\mathbf{V}$. Then, by Corollary 6.3.11, there exist finite Boolean sub-algebras $\mathcal{B}_1$ and $\mathcal{B}_2$ of $\mathcal{V}(A)$ such that $\mathsf{R}_1 = \mathsf{R}(\mathcal{B}_1)$ and $\mathsf{R}_2 = \mathsf{R}(\mathcal{B}_2)$.

Let $\mathcal{B}$ be the Boolean algebra generated by $\mathcal{B}_1$ and $\mathcal{B}_2$, then $\mathcal{B}$ is a subalgebra of $\mathcal{V}(A)$ and hence $\mathsf{R}(\mathcal{B}) \in \mathcal{I}_A(\mathbf{V})$ recognises both $\mathcal{B}_1$ and $\mathcal{B}_2$. It follows that both R_1 and R_2 divide $\mathsf{R}(\mathcal{B})$. Thus $\mathsf{R}(\mathcal{B})$ is an upper bound and $\mathcal{I}_A(\mathbf{V})$ is directed. ∎

To adapt to the usual notation for projective limits, we let $\mathcal{I}_A(\mathbf{V}) = (\mathsf{R})_{i\in I}$, where I is some appropriate index set, such that $i \leq j$ if and only if $\mathsf{R}_i \prec \mathsf{R}_j$. The (uniquely determined) factoring morphisms are denoted by $\Psi_{ij}\colon \mathsf{R}_j \to \mathsf{R}_i$.

6.4.2 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps, then the set $\mathcal{I}_A(\mathbf{V})$ forms a projective system.

Proof. Let $i, j \in I$ with $i \leq j$. The connection morphisms are the morphisms $\Psi_{ij}\colon \mathsf{R}_j \to \mathsf{R}_i$.

Obviously, for each $i \in I$, $\Psi_{ii}\colon \mathsf{R}_i \to \mathsf{R}_i$ is the identity.

Let $i, j, k \in I$ such that $i \leq j \leq k$. Since $i \leq k$ implies that there exists exists exactly one morphism Ψ_{ik} between R_k and R_i, we conclude that $\Psi_{ij} \circ \Psi_{jk} = \Psi_{ik}$. ∎

Just like the projective limit of a variety of finite monoids is in general not a finite monoid anymore, the projective limit of an inherent $\mathcal{C}$-stream does also in general not exist in the category of typed stamps:

Let $\mathcal{V}$ be the *all*-stream of all regular languages and $\mathbf{V}$ the corresponding *all*-stream of typed stamps – this is precisely the stream considered in Example 6.2.3. Then for any two words $u, v \in A^*$, there exists a typed stamps $\mathsf{R} \in \mathcal{I}(\mathbf{V})$ such that $\iota_{\mathsf{R}}(u) \neq \iota_{\mathsf{R}}(v)$, but there exists no typed stamp T such that $\iota_{\mathsf{T}}(u) \neq \iota_{\mathsf{T}}(v)$ for all $u, v \in A^*$. Hence there exists no typed stamp T with projection morphisms $\Pi_{\mathsf{R}}\colon \mathsf{T} \to \mathsf{R}$ for each $\mathsf{R} \in \mathcal{I}_A(\mathbf{V})$. Which proves that $\mathcal{I}_A(\mathbf{V})$ does not have a projective limit in the category of typed stamps.

Projective Limits and Dense Stamps

In order to characterise in which category the projective limit of $\mathcal{I}_A(\mathbf{V})$ exists, we briefly consider typed stamps as topological objects and derive therefrom a natural extension to a more general category.

Let $\mathsf{R} = (A, \mu, M, p, X) \in \mathcal{I}_A(\mathbf{V})$, then X is the Stone space of $L(\mathsf{R})$ and as such a finite, discrete topological space and $p\colon M \to X$ is a map with dense image, since it is surjective. Moreover, if also $\mathsf{S} \in \mathcal{I}_A(\mathbf{V})$ and $\Psi\colon \mathsf{R} \to \mathsf{S}$ is a morphism with $\Psi = (\mathrm{id}_{A^*}, \phi, f)$, then f is a continuous function between discrete topological spaces.

From these observation, we consider the projective limit in the category of *dense stamps*.

6.4.3 Definition

A dense stamp R is a tuple (A, μ, M, p, X), where A is an alphabet, M a monoid, $\mu\colon A^* \to M$ a monoid morphism, X a Stone space and $p\colon M \to X$ a map with dense image.

6.4.4 Definition

Let $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (B, \nu, N, q, Y)$ be dense stamps. A morphism of dense stamps $\Psi\colon \mathsf{R} \to \mathsf{S}$ is a triple $\Psi = (h, \phi, f)$ where $h\colon A^* \to B^*$ and $\phi\colon \mu(A^*) \to N$ are monoid morphisms and $f\colon X \to Y$ is a continuous function such that the diagram

$$\begin{array}{ccccc} A^* & \xrightarrow{\mu} & \mu(A^*) & \xrightarrow{p} & X \\ {\scriptstyle h}\downarrow & & {\scriptstyle \phi}\downarrow & & {\scriptstyle f}\downarrow \\ B^* & \xrightarrow{\nu} & \nu(A^*) & \xrightarrow{q} & Y \end{array}$$

commutes.

Observe that if $\mathsf{R} = (A, \mu, M, p, X)$ is a typed stamp, then X is a discrete space and hence p as a surjective function has dense image. In addition, every morphism of typed stamps is a morphism of dense stamps. Hence, each typed stamp in $\mathcal{I}_A(\mathbf{V})$ also is a dense stamp and we prove that in this category, the projective limit exists and exhibits some interesting properties.

Recall that, if $(X_i)_{i\in I}$ is a family of finite discrete spaces, we consider $\prod_{i\in I} X_i$ to be equipped with the product topology and any subset $X \subseteq \prod_{i\in I} X_i$ is equipped with the subspace topology.

6.4.5 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps. Then the projective limit of the inherent class $\mathcal{I}_A(\mathbf{V}) = (\mathsf{R}_i)_{i\in I}$ with $\mathsf{R}_i = (A, \mu_i, M_i, p_i, X_i)$ is the dense stamp

$$\widehat{\mathsf{R}}_A(\mathbf{V}) = (A, \mu, M, p, X)$$

where

- $M \subseteq \prod_{i\in I} M_i$ is the set $\{(\mu_i(w))_{i\in I} \mid w \in A^*\}$,
- the map μ sends w to $(\mu_i(w))_{i\in I}$,
- $X \subseteq \prod_{i\in I} X_i$ is the topological closure of the set $\{(\iota_{\mathsf{R}_i}(w))_{i\in I} \mid w \in A^*\}$,
- for each $(m_i)_{i\in I} \in M$, the map p is defined by $p((m_i)_{i\in I}) = (p_i(m_i))_{i\in I}$

Proof. We first observe that the defined object is indeed a dense stamp: Since X is a closed subset of a Stone space, it is – equipped with the subspace topology – also a Stone space. The map p is dense, since each $(m_i)_{i\in I}$ is equal to $(\mu_i(w))_{i\in I}$ for some $w \in A^*$ and hence $p(M) = \{(\iota_{\mathsf{R}_i}(w))_{i\in I} \mid w \in A^*\}$, which is dense in X by definition, as X is the topological closure of that set.

We now show that for each typed stamp $\mathsf{R}_i \in \mathcal{I}_A(\mathbf{V})$, there exists a morphism of dense stamps $\Pi_i \colon \widehat{\mathsf{R}}_A(\mathbf{V}) \to \mathsf{R}_i$ commuting with the connection morphisms.

For each $i \in I$, let

$$\begin{aligned} \pi_{M_i} \colon M &\to M_i \\ (m_i)_{i\in I} &\mapsto m_i \end{aligned}$$

be the projection onto the ith component and similarly let $\pi_{X_i} \colon \iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(A^*) \to X_i$ be the map sending $(x_i)_{i\in I} \in \iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(A^*)$ to ith component x_i.

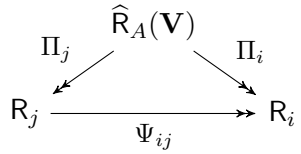
By definition, $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(A^*)$ is dense in X. Hence there exists a unique continuous extension $\widehat{\pi_{X_i}} \colon X \to X_i$ and the projection morphisms $\Pi_i \colon \widehat{\mathsf{R}}_A(\mathbf{V}) \to \mathsf{R}_i$ are given by the triples

$$\Pi_i = (\mathrm{id}_{A^*}, \pi_{M_i}, \widehat{\pi_{X_i}}).$$

Let $\mathsf{R}_i, \mathsf{R}_j \in \mathcal{I}_A(\mathbf{V})$ such that $j \geq i$ and let $\Psi_{ji} \colon \mathsf{R}_j \to \mathsf{R}_i$ with $\Psi_{ij} = (\mathrm{id}_{A^*}, \psi_{ji}, f_{ji})$ denote the connection morphism. Then the diagram

$$\begin{array}{ccccc} A^* & \xrightarrow{\mu_j} & M_j & \xrightarrow{p_j} & X_j \\ & \searrow^{\mu_i} & \downarrow{\scriptstyle \psi_{ij}} & & \downarrow{\scriptstyle f_{ij}} \\ & & M_i & \xrightarrow{p_i} & X_i \end{array}$$

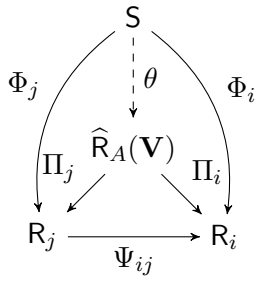
commutes. From this diagram, it is not hard to verify that the projections commute with the connection morphisms in the sense, that the diagram below also commutes.



It now remains to be shown that $\widehat{\mathsf{R}}_A(\mathbf{V})$ together with the projection morphisms Π_i is the projective limit. We prove that it satisfies the universal property.

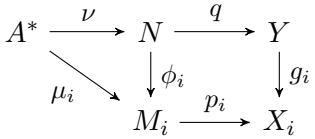
Let $\mathsf{S} = (A, \nu, N, q, Y)$ be a dense stamp such that for each $i \in I$ there exist morphisms $\Phi_i \colon \mathsf{S} \to \mathsf{R}_i$ with $\Phi_i = (\mathrm{id}_{A^*}, \phi_i, g_i)$ that commute with the connection morphisms.

We construct a morphism of dense stamps $\theta \colon \mathsf{S} \to \widehat{\mathsf{R}}_A(\mathbf{V})$, such that the diagram



commutes.

Since Φ_i is a morphism the diagram below ensures that for any $n \in \nu(A^*) \subseteq N$, $(\phi_i(n))_{i\in I}$ is an element of M and also that for each $y \in \iota_{\mathsf{S}}(A^*)$, $(g_i(y))_{i\in I}$ is an element of X.



Define $\varphi \colon \nu(A^*) \to M$ as the map sending n to $(\phi_i(n))_{i\in I}$. This, by the previous observation is well-defined. From the diagram above, we obtain in particular that

$$\mu(w) = (\mu_i(w))_{i\in I} = (\phi_i \circ \nu(w))_{i\in I} = \varphi \circ \nu(w).$$

Hence $\varphi(\nu(A^*)) = M$.

Since each of the maps $g_i \colon Y \to X_i$ are continuous, also the map $f \colon Y \to \prod_{i\in I} X_i$ which sends y to $(g_i(y))_{i\in I}$ is continuous. This is a straight forward consequence of $\prod_{i\in I} X_i$ being equipped with the product topology. In the same fashion as for φ, we obtain

$$\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(w) = f \circ \iota_{\mathsf{S}}(w),$$

which implies that $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(A^*) = f(\iota_{\mathsf{S}}(A^*))$ and since $\iota_{\mathsf{S}}(A^*)$ is dense in Y and f continuous

$$f(Y) = f(\overline{\iota_{\mathsf{S}}(A^*)}) = \overline{f(\iota_{\mathsf{S}}(A^*))} = \overline{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(A^*)} = X.$$

Hence f is a continuous map from Y to X and summarising, $\theta = (\mathrm{id}_{A^*}, \varphi, f)$ is a morphism of dense stamps from S to $\widehat{\mathsf{R}}_A(\mathbf{V})$. ■

If $\mathbf{V}$ is a $\mathcal{C}$-stream of typed stamps, then we call $\widehat{\mathsf{R}}_A(\mathbf{V})$ the dense pro-$\mathbf{V}$ stamp over A.

6.5 Properties of the Dense Pro-$\mathbf{V}$ Stamp

While the definition through profinite limits is rather technical, the dense pro-$\mathbf{V}$ stamp enjoys some useful properties.

Notation: In order to avoid redundancies, we fix some notation for this section. Let $\mathbf{V}$ be a stream of typed stamps, then

$$\widehat{\mathsf{R}}_A(\mathbf{V}) = (A, \mu, M, p, X).$$

The inherent class $\mathcal{I}_A(\mathbf{V})$ is the family $(\mathsf{R}_i)_{i\in I}$ where

$$\mathsf{R}_i = (A, \mu_i, M_i, p_i, X_i).$$

Denote by $\Pi_i \colon \widehat{\mathsf{R}}_A(\mathbf{V}) \to \mathsf{R}_i$ the projections, with $\Pi_i = (\mathrm{id}_{A^*}, \pi_{M_i}, \widehat{\pi_{X_i}})$, where $\widehat{\pi_{X_i}}$ is the unique continuous extension of the map sending $(\iota_{\mathsf{R}_i}(w))_{i\in I} \in X$ to $\iota_{\mathsf{R}_i}(w) \in X_i$.

6.5.1 Proposition

Let $\mathbf{V}$ be a stream of typed stamps, $\mathcal{V}$ be the associated stream of languages and $\widehat{\mathsf{R}}_A(\mathbf{V})$ be the dense pro-$\mathbf{V}$ stamp over A. Then

1. M is the syntactic monoid of $\mathcal{V}(A)$ and
2. μ is the syntactic morphism of $\mathcal{V}(A)$.

Proof. We show that for any $u, v \in A^*$, $\mu(u) = \mu(v)$ implies $u \sim_{\mathcal{V}(A)} v$, where $\sim_{\mathcal{V}(A)}$ for the infinite Boolean algebra $\mathcal{V}(A)$ is defined in the same way as the syntactic congruence for finite Boolean algebras. Let $x, y \in A^*$. Then

$$\begin{aligned}
\mu(xuy) = \mu(xvy) &\Rightarrow \forall i \in I\ \mu_i(xuy) = \mu_i(xvy) \\
&\Rightarrow \forall i \in I\ \forall C \subseteq X_i\ (\iota_{\mathsf{R}_i}(xuy) \in C \Leftrightarrow \iota_{\mathsf{R}_i}(xvy) \in C) \\
&\Rightarrow \forall i \in I\ \forall C \subseteq X_i\ (xuy \in \iota_{\mathsf{R}_i}^{-1}(C) \Leftrightarrow xvy \in \iota_{\mathsf{R}_i}^{-1}(C)) \\
&\Rightarrow \forall L \in \mathcal{V}(A)\ (xuy \in L \Leftrightarrow xvy \in L)
\end{aligned}$$

since the languages recognised by members in $\mathcal{I}_A(\mathbf{V})$ are precisely the languages in $\mathcal{V}(A)$.

For the converse direction assume that $\mu(u) \neq \mu(v)$. Then there exists an $i \in I$ and a typed stamp R_i such that $\mu_i(u) \neq \mu_i(v)$. By Corollary 6.3.11, R_i is the syntactic

typed stamp of some finite Boolean algebra $\mathcal{B} \subseteq \mathcal{V}(A)$ and hence there exist $x, y \in A^*$ and a language $L \in \mathcal{B} \subseteq \mathcal{V}(A)$ such that

$$xuy \in L \text{ and } xvy \notin L.$$

This proves that $u \not\sim_{\mathcal{V}(A)} v$. Hence μ is the syntactic morphism of $\mathcal{V}(A)$ and since M is the image of μ, it is (isomorphic to) the syntactic monoid of $\mathcal{V}(A)$. ∎

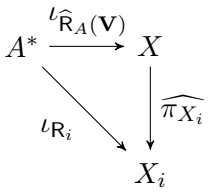
6.5.2 Lemma

Let $\mathbf{V}$ be a stream of typed stamps, $\mathcal{V}$ be the associated stream of languages and $\widehat{\mathsf{R}}_A(\mathbf{V})$ be the dense pro-$\mathbf{V}$ stamp over A. For $L \in \mathcal{V}(A)$ let

$$\widehat{L} = \overline{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(L)}.$$

Then $\widehat{L}$ is clopen in X and $L = \iota^{-1}_{\widehat{\mathsf{R}}_A(\mathbf{V})}(\widehat{L})$.

Proof. Let $L \in \mathcal{V}(A)$. Then there exists an $i \in I$ such that R_i recognises L and the diagram



commutes. From the continuity of $\widehat{\pi_{X_i}}$ and the fact that X_i is discrete as a space, we obtain

$$\begin{aligned}
\widehat{\pi_{X_i}}^{-1}(\iota_{\mathsf{R}_i}(L)) &= \widehat{\pi_{X_i}}^{-1}(\overline{\iota_{\mathsf{R}_i}(L)}) \\
&= \overline{\widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L)} \\
&= \overline{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(L)} \\
&= \widehat{L}
\end{aligned}$$

Since X_i is a discrete space, $\iota_{\mathsf{R}_i}(L)$ is clopen and hence also $\widehat{L}$. In particular, since $L = \iota^{-1}_{\mathsf{R}_i}(\iota_{\mathsf{R}_i}(L))$, we obtain by applying $\iota^{-1}_{\widehat{\mathsf{R}}_A(\mathbf{V})}$ to both sides of the previous equality and using commutativity of the above diagram

$$\iota^{-1}_{\widehat{\mathsf{R}}_A(\mathbf{V})}(\widehat{L}) = \iota^{-1}_{\widehat{\mathsf{R}}_A(\mathbf{V})}(\widehat{\pi_{X_i}}^{-1}(\iota_{\mathsf{R}_i}(L))) = \iota^{-1}_{\mathsf{R}_i}(\iota_{\mathsf{R}_i}(L)) = L.$$

∎

Recall that for any topological space X, the set $\mathrm{Clopen}(X)$ is the set of clopens of X.

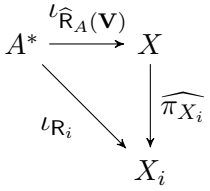
6.5.3 Lemma

Let $\mathbf{V}$ be a stream of typed stamps, $\mathcal{V}$ be the associated stream of languages and $\widehat{\mathsf{R}}_A(\mathbf{V})$ be the dense pro-$\mathbf{V}$ stamp over A. The map

$$\begin{aligned} \sigma \colon \mathcal{V}(A) &\to \mathrm{Clopen}(X) \\ L &\mapsto \widehat{L}. \end{aligned}$$

is an isomorphism of Boolean algebras.

Proof. We first prove that σ is a morphism of Boolean algebras. For any language $L \in \mathcal{V}(A)$ and typed stamp $\mathsf{R}_i \in \mathcal{I}_A(\mathbf{V})$ recognising L, the diagram



commutes and

$$\widehat{L} = \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L).$$

Now let $L, K \in \mathcal{V}(A)$ and $\mathsf{R}_i \in \mathcal{I}_A(\mathbf{V})$ be a typed stamp recognising both. Then ι_{R_i} induces a morphism of Boolean algebras from $L(\mathsf{R}_i)$ to the powerset of X_i and hence

$$\begin{aligned} \widehat{L \cap K} &= \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L \cap K) \\ &= \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L) \cap \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(K) \\ &= \widehat{L} \cap \widehat{K}. \end{aligned}$$

The cases for intersection and complementation are completely analogous. It remains to be shown that σ is in fact an isomorphism of Boolean algebras.

That σ is surjective follows immediately from X being equipped with the product topology of the sets X_i and hence each clopen is a Boolean combination of sets $\widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L)$, where L is some language in $\mathcal{V}(A)$.

To prove that σ is injective, let $L, K \in \mathcal{V}(A)$ such that $L \neq K$ and let $\mathsf{R}_i \in \mathcal{I}_A(\mathbf{V})$ be a typed stamp recognising both L and K. Then $\iota_{\mathsf{R}_i}(L) \neq \iota_{\mathsf{R}_i}(K)$, which implies

$$\widehat{\pi_{X_i}}(\widehat{L}) = \iota_{\mathsf{R}_i}(L) \neq \iota_{\mathsf{R}_i}(K) = \widehat{\pi_{X_i}}(\widehat{K}).$$

Since $\widehat{\pi_{X_i}}$ is surjective, we conclude by applying $\widehat{\pi_{X_i}}^{-1}$ to each of the (in-)equalities above that

$$\widehat{L} = \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(L) \neq \widehat{\pi_{X_i}}^{-1} \circ \iota_{\mathsf{R}_i}(K) = \widehat{K}.$$

Hence σ is an isomorphism of Boolean algebras.

■

This already suffices to see that X is the Stone space of $\mathcal{V}(A)$, since Stone spaces are uniquely determined by their clopens. The next proposition also characterises the map $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}$ more precisely.

6.5.4 Proposition

Let $\mathbf{V}$ be a stream of typed stamps, $\mathcal{V}$ be the associated stream of languages and $\widehat{\mathsf{R}}_A(\mathbf{V})$ be the dense pro-$\mathbf{V}$ stamp over A. Then

1. X is isomorphic to the Stone space of $\mathcal{V}(A)$,
2. the map $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}$ is (modulo isomorphism) the canonical inclusion of A^* in the Stone space of $\mathcal{V}(A)$.

Proof. Since $\widehat{\mathsf{R}}_A(\mathbf{V})$ is a dense stamp, X is a Stone space and since Stone spaces are uniquely determined by their clopens, it suffices to show that there exists an isomorphism of Boolean algebras between the clopens of X and the clopens of $X_{\mathcal{V}(A)}$.

Recall that we may include A^* in $X_{\mathcal{V}(A)}$ via the map

$$\begin{aligned}\iota\colon A^* &\to X_{\mathcal{V}(A)}\\ w &\mapsto \{w \in \mathcal{V}(A) \mid w \in L\}\end{aligned}$$

and hence any language $L \in \mathcal{V}(A)$ has an embedding $\iota(L) \subseteq X_{\mathcal{V}(A)}$. Moreover, remember that there is a one-to-one correspondence between clopens of $X_{\mathcal{V}(A)}$ and languages in $\mathcal{V}(A)$ via the map sending $L \in \mathcal{V}(A)$ to the closure of its inclusion $\overline{L} = \{\mu \in X_{\mathcal{V}(A)} \mid L \in \mu\}$. Hence we identify each clopen of $X_{\mathcal{V}(A)}$ with such a set.

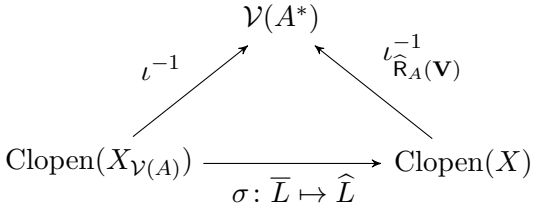
By Lemma 6.5.2, for each $L \in \mathcal{V}(A)$, the set $\widehat{L} = \overline{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(L)}$ is clopen and we let

$$\begin{aligned}\sigma\colon \mathrm{Clopen}(X_{\mathcal{V}(A)}) &\to \mathrm{Clopen}(X)\\ \overline{L} &\mapsto \widehat{L}.\end{aligned}$$

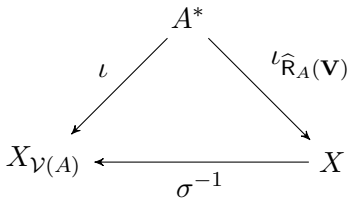
where $\mathrm{Clopen}(X_{\mathcal{V}(A)})$ is the set of clopens of $X_{\mathcal{V}(A)}$ and respectively $\mathrm{Clopen}(X)$ the set of clopens of X.

By Lemma 6.5.3, the map sending $L \in \mathcal{V}(A)$ to $\widehat{L}$ is an isomorphism of Boolean algebras and since $L \mapsto \overline{L}$ also is an isomorphism, so is σ, which makes X the Stone spaces of $\mathcal{V}(A)$ and proves the first claim.

For the second claim, observe that for $L \in \mathcal{V}(A)$, we have $\iota^{-1}(\overline{L}) = L$ and by Lemma 6.5.2, $\iota^{-1}_{\widehat{\mathsf{R}}_A(\mathbf{V})}(\widehat{L}) = L$. Hence the diagram



commutes and by duality also the diagram



commutes. Which proves that $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}$ is the canonical inclusion of A^* in $X_{\mathcal{V}(A)}$ modulo isomorphism. ∎

Recognition on Dense Stamps

Regarding the previous findings, we may now generalise concepts like recognition to dense stamps. To be able to talk about a dense stamp recognising some set of languages makes the terminology a tad more light-weight in some places.

6.5.5 Definition

A dense stamp $\mathsf{R} = (A, \mu, M, p, X)$ recognises some language $L \subseteq A^*$, if there exists a clopen $C \subseteq X$ such that $L = \iota_{\mathsf{R}}^{-1}(C)$. We write $L(\mathsf{R})$ for the set of all languages recognised by R.

Hence, by Lemma 6.5.2, we obtain the following Corollary.

6.5.6 Corollary

Let $\mathbf{V}$ be a stream of typed stamps and $\mathcal{V}$ its associated stream of languages. Then a language is in $\mathcal{V}(A)$ if and only if it is recognised by $\widehat{\mathsf{R}}_A(\mathbf{V})$.

Similarly to typed stamps, we may define the notion of syntactic typed stamp of a Boolean algebra.

6.5.7 Definition

Let $\mathcal{B}$ be a Boolean algebra of languages over A^*. Then the syntactic dense stamp is the tuple

$$(A^*, \eta_{\mathcal{B}}, M_{\mathcal{B}}, p_{\mathcal{B}}, X_{\mathcal{B}})$$

where $M_{\mathcal{B}}$ is the syntactic monoid of $\mathcal{B}$, $\eta_{\mathcal{B}}$ is the syntactic morphism, $X_{\mathcal{B}}$ the Stone space of $\mathcal{B}$ and the map $p_{\mathcal{B}}$ uniquely determined by the embedding

$$\iota_{\mathsf{R}}(w) = \{\mu \in X_{\mathcal{B}} \mid w \in \mu\}.$$

As such, it is clear that $\widehat{\mathsf{R}}_A(\mathbf{V})$ is the syntactic dense stamp of $\mathcal{V}(A)$ and we obtain the following useful properties, that hold just like for typed stamps.

6.5.8 Definition

Let $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$ be two dense stamps. Then S factors through R if there exists a morphism of dense stamps $\Psi\colon \mathsf{R} \to \mathsf{S}$ such that $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ where ϕ and f are quotient maps. We say that R is a trivial extension of S, if S factors through R and f is an isomorphism.

6.5.9 Proposition

A dense stamp R recognises a Boolean algebra $\mathcal{B}$ if and only if the syntactic dense stamp factors through R.

Proof. Let $\mathcal{B}$ be a Boolean algebra of languages over A^* and let $\mathsf{R}(\mathcal{B})$ be its syntactic dense stamp. Moreover let $\mathsf{R} = (A, \mu, M, p, X)$ be a dense stamp recognising $\mathcal{B}$. We claim that there exists a quotient morphism $\phi\colon M \to M_{\mathcal{B}}$ and a continuous function $f\colon X \to X_{\mathcal{B}}$ such that $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ is a morphism $\Psi\colon \mathsf{R} \to \mathsf{R}(\mathcal{B})$.

We prove that $\eta_{\mathcal{B}}$ factors through μ. Let $u, v \in A^*$ such that $\mu(u) = \mu(v)$, which implies $\mu(xuy) = \mu(xvy)$ for all $x, y \in A^*$ and hence also $\iota_{\mathsf{R}}(xuy) = \iota_{\mathsf{R}}(xvy)$. We obtain, since R recognises $\mathcal{B}$, that for each $L \in \mathcal{B}$

$$xuy \in L \Leftrightarrow xvy \in L.$$

Hence μ factors through the syntactic morphism $\eta_{\mathcal{B}}$ and there exists a morphism $\phi\colon M \to M_{\mathcal{B}}$ such that $\eta_{\mathcal{B}} = \phi \circ \mu$.

Since X is a Stone space and R recognises $\mathcal{B}$, X has a Boolean subalgebra of clopens isomorphic to $\mathcal{B}$ and hence there exists a continuous quotient $f\colon X \to X_{\mathcal{B}}$ such that $f \circ \iota_{\mathsf{R}} = \iota_{\mathsf{R}(\mathcal{B})}$. This implies that $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ is a morphism, which proves one direction of the claim.

For the converse direction, assume that $\mathsf{R}(\mathcal{B})$ factors through R, that is there exists a morphism $\Psi = (\mathrm{id}_{A^*}, \phi, f)$ with $\Psi\colon \mathsf{R} \to \mathsf{R}(\mathcal{B})$. Let $L \in \mathcal{B}$, then

$$L = \iota^{-1}_{\mathsf{R}(\mathcal{B})}(\overline{\iota_{\mathsf{R}(\mathcal{B})}(L)}) = \iota^{-1}_{\mathsf{R}}(f^{-1}(\overline{\iota_{\mathsf{R}(\mathcal{B})}(L)})),$$

which proves that R recognises L. ∎

6.5.10 Proposition

Let R and S be two dense stamps. Then $L(\mathsf{R}) = L(\mathsf{S})$ if and only if there exists a dense stamp T such that R and S are trivial extensions of T.

Proof. If there exists a dense stamp T such that S and R are trivial extensions of T, then it follows immediately from the fact that there exist morphisms $\Phi\colon \mathsf{S} \to \mathsf{T}$ and

$\Psi\colon \mathsf{R} \to \mathsf{T}$ for which the last component is an isomorphism, that they recognise the same languages.

For the converse direction, let $\mathcal{B} = L(\mathsf{R}) = L(\mathsf{S})$. We prove that R is a trivial extension of the syntactic dense stamp $\mathsf{R}(\mathcal{B})$. By Proposition 6.5.9, $\mathsf{R}(\mathcal{B})$ factors through R where $\Psi\colon \mathsf{R} \to \mathsf{R}(\mathcal{B})$ is the morphism with $\Psi = (\mathrm{id}_{A^*}, \phi, f)$. Since both recognise the same languages, their Stone spaces are isomorphic and hence f must be an isomorphism.

Replacing R by S gives the same result. ∎

6.6 Concrete Dense Stamps Calculated

We now use the previous characterisations to determine the dense pro-$\mathbf{V}$ stamp for some concrete $\mathcal{C}$-streams.

The Finite Co-Finite Algebra

The set of all languages over A^* that are either finite or have finite complement form a Boolean algebra. If we consider any non-erasing morphism $\varphi\colon B^* \to A^*$, then if L is either finite or has finite complement also $\varphi^{-1}(L)$ is finite or has finite complement.

Remark: Observe that this is in general not the case, if φ may erase letters. If $\varphi\colon \{a,b\}^* \to \{b\}^*$ sends a to λ and b to b, then both $\varphi^{-1}(\{b\})$ and $\varphi^{-1}(\{b\})^c$ are infinite.

Hence the class $\mathcal{V}_{\text{Co-Fin}}$ associating to each alphabet A the Boolean algebra of all finite or co-finite languages, is an *ne*-stream and as such has an associated *ne*-stream of typed stamps $\mathbf{V}_{\text{Co-Fin}}$.

We will now have a closer look at the dense pro-$\mathbf{V}_{\text{Co-Fin}}$ stamp over A

$$\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}}) = (A, \mu, M, p, X).$$

Observe that for each $w \in A^*$, the singleton language $\{w\}$ is in $\mathcal{V}_{\text{Co-Fin}}$ and hence M, as the syntactic monoid of $\mathcal{V}_{\text{Co-Fin}}$ is equal to A^* and μ equal to id_{A^*}. Thus

$$\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}}) = (A^*, \mathrm{id}_{A^*}, A^*, p, X).$$

A bit more interesting in this case is X. We continue by determining a suitable representation for X, which is the Stone space of $\mathcal{V}_{\text{Co-Fin}}(A)$.

An ultrafilter over $\mathcal{V}_{\text{Co-Fin}}$ contains either a singleton $\{w\}$ or, if it does not, it is the ultrafilter consisting entirely of co-finite languages. We represent that ultrafilter by ∞ and all singleton-containing filters by their respective singleton w.

Hence X as a set is equal to $A^* \cup \{\infty\}$. Since the topology on X is generated by the sets $\overline{L} = \{\mu \in X_{\mathcal{V}_{\text{Co-Fin}}} \mid L \in \mu\}$ and $\mathcal{V}_{\text{Co-Fin}}(A)$ is generated by the finite and

co-finite sets, we obtain as generators for the topology

$$\overline{L} = \begin{cases} L & \text{if } L \text{ is finite} \\ L \cup \{\infty\} & \text{if } L \text{ is co-finite.} \end{cases}$$

One may recognise this as the one-point-compactification of A^* as a discrete space.

Since $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}})}(w) = \{L \in \mathcal{V}_{\text{Co-Fin}}(A) \mid w \in L\} = w$, the map $p\colon A^* \to A^* \cup \{\infty\}$ is the inclusion i_{A^*} and hence

$$\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}}) = (A^*, \mathrm{id}_{A^*}, A^*, i_{A^*}, A^* \cup \{\infty\}).$$

In this case, since the Boolean algebra itself carries little algebraic information, neither does the syntactic monoid and all the information about it is contained in the topology on X. Observe that we cannot recognise any languages apart from the finite or co-finite ones due to the topology, where the only clopen sets are the finite ones or the co-finite ones united with ∞.

A Non-Regular Boolean algebra

As the title indicates, we now consider a stream of languages which consists in contrast to the previous examples, of non-regular languages.

To illustrate that the monoid-component is not always negligible, as in the previous Example, we will consider a stream of languages for which the topological component of its syntactic dense stamp is essentially the same as in the finite co-finite case, but the monoid differs significantly.

We consider the *lp*-stream of languages $\mathcal{V}_\Lambda$ generated by the set

$$\Lambda := \{a^n b^n \mid a, b \in A, n \in \mathbb{N}\}$$

and the sets $\overline{n} := \{a^n b^n \mid a, b \in A\}$ for $n \in \mathbb{N}$.

Remark: To prevent confusion in the first place, since it may be overlooked: Λ contains both $a^k b^k$ and $b^k a^k$ for $k \in \mathbb{N}$.

Consider the Boolean algebra those sets generate: It is immediately clear that it contains the finite co-finite algebra relative to Λ, that is all sets that are either finite subsets of Λ or complements thereof. The remaining sets are unions of the complement of Λ and some element of the finite co-finite algebra relative to Λ.

Since we are only interested in the *lp*-stream generated by Λ and $\overline{n}$ and length-preserving morphisms only replace letters, we may quickly verify that inverse morphisms contribute no new elements to the Boolean algebra.

Having clarified the structure of the *lp*-stream, we now consider the dense pro-$\mathbf{V}_\Lambda$ stamp

$$\widehat{\mathsf{R}}_A(\mathbf{V}_\Lambda) = (A, \mu, M, p, X).$$

We identify the syntactic monoid of $\mathcal{V}_\Lambda(A)$: Observe that two words of the form $a^i b^j$ and $c^n d^m$, where $a, b, c, d \in A$ and $i, j, n, m \in \mathbb{N}$ can not syntactically be separated

by some language in $\mathcal{V}_\Lambda(A)$ if and only if $i-j=n-m$ and $a=c, b=d$, since for all $x,y\in A^*$, the words xa^ib^jy and xc^nd^my are either in the same set $\overline{l}$ for some $l\in\mathbb{N}$ or elements of Λ^c. If w is some word that is not of the form a^ib^j, then $xwy\in\Lambda^c$ for all $x,y\in A^*$.

Hence the syntactic monoid of $\mathcal{V}_\Lambda(A)$ is the set

$$M=\{[a^ib^j]\mid a,b\in A \text{ and } i,j\in\mathbb{N}\}\cup\{0\}$$

where $[a^ib^j]=[c^nd^m]$, if and only if $i-j=n-m$ and $a=c, b=d$. The multiplication is given by

$$[a^ib^j]\cdot[c^nd^m]=\begin{cases}[a^ib^{j+m}] & \text{if } n=0 \text{ and } d=b\\ [a^{i+n}b^m] & \text{if } j=0 \text{ and } a=c\\ 0 & \text{otherwise.}\end{cases}$$

where 0 is an absorbing element.

The syntactic morphism μ sends any word of the form a^ib^j to $[a^ib^j]$ and all other words to 0.

We now consider the Stone space X of $\mathcal{V}_\Lambda$: Just as in the previous example, an ultrafilter contains either a set $\overline{n}$ or if it does not, it may be one out of two remaining ultrafilters:

Observe that each ultrafilter contains either Λ or Λ^c, hence in the case that it contains Λ, it must be the co-finite filter which consists of all co-finite subsets of Λ. If it contains Λ^c, it is the ultrafilter which consists of all supersets of Λ^c in $\mathcal{V}_\Lambda$.

We identify each ultrafilter containing a singleton with the corresponding natural number n, the co-finite ultrafilter is again denoted by ∞ and the ultrafilter containing Λ^c by -1. Thus

$$X=\mathbb{N}\cup\{-1,\infty\}.$$

Again, the topology is generated by the sets $\overline{L}=\{\mu\in X_{\mathcal{V}_\Lambda(A)}\mid L\in\mu\}$ and hence we obtain for the generators of the Boolean algebra (and thus the generators of the topology):

$$\overline{L}=\begin{cases}P_L & \text{if } L\subseteq\Lambda \text{ and } L \text{ is finite}\\ P_L\cup\{\infty\} & \text{if } L\subseteq\Lambda \text{ and } L \text{ is co-finite}\\ -1 & \text{if } \Lambda^c\subseteq L.\end{cases}$$

where P_L is the set of all $n\in\mathbb{N}$ such that $a^nb^n\in L$.

One may observe that the topology itself contains little more than the finite co-finite information that already the previous example contained. The algebraic structure of the language is maintained in the monoid, not in the topological space.

6.7 Equations and Typed Stamps

In this section, we give a Reiterman-like characterisation for streams of typed stamps. Using duality and in particular ultrafilters, we start by deriving an interpretation of

ultrafilter equations on typed stamps.

Consider a typed stamp $\mathsf{R} = (A, \mu, M, p, X)$. Then, since X is a finite discrete space and hence also compact, the map $\iota_{\mathsf{R}} \colon A^* \to X$ has a unique continuous extension

$$\widehat{\iota_{\mathsf{R}}} \colon \beta(A^*) \to X.$$

Recall that for any ultrafilter $\gamma \in \beta(A^*)$, the extension maps γ to some element $x \in X$ if and only if $\iota_{\mathsf{R}}^{-1}(x) \in \gamma$.

6.7.1 Definition

Let $\mathsf{R} = (A, \mu, M, p, X)$ be a typed stamp.

- Let $\gamma_1, \gamma_2 \in \beta(A^*)$. We say that R satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$ if and only if $\widehat{\iota_{\mathsf{R}}}(\gamma_1) = \widehat{\iota_{\mathsf{R}}}(\gamma_2)$.
- Let B be an alphabet with $\gamma_1, \gamma_2 \in \beta(B^*)$. We say that R satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to the morphism $h \colon B^* \to A^*$, if and only if R satisfies $[\beta h(\gamma_1) \leftrightarrow \beta h(\gamma_2)]$.

6.7.2 Lemma

Let R be a typed stamp over the alphabet A, let B be a second alphabet, $h \colon B^* \to A^*$ a morphism and $\gamma_1, \gamma_2 \in \beta(B^*)$. Then the following statements are equivalent:

1. $\mathsf{R} \cdot h$ satisfies $[\gamma_1 \leftrightarrow \gamma_2]$.
2. R satisfies $[\gamma_2 \leftrightarrow \gamma_2]$ with respect to h.

Proof. Let $\mathsf{R} = (A, \mu, M, p, X)$ be a typed stamp and $h \colon B^* \to A^*$ be a morphism. We show that $\widehat{\iota_{\mathsf{R}\cdot h}} = \widehat{\iota_{\mathsf{R}}} \circ \beta h$. Let $\gamma \in \beta(B^*)$ and $x \in X$. Then

$$\begin{aligned}
\widehat{\iota_{\mathsf{R}\cdot h}}(\gamma) = x &\Leftrightarrow \iota_{\mathsf{R}\cdot h}^{-1}(x) \in \gamma \\
&\Leftrightarrow (\iota_{\mathsf{R}} \circ h)^{-1}(x) \in \gamma \\
&\Leftrightarrow h^{-1}(\iota_{\mathsf{R}}^{-1}(x)) \in \gamma \\
&\Leftrightarrow \iota_{\mathsf{R}}^{-1}(x) \in \beta h(\gamma) \\
&\Leftrightarrow \widehat{\iota_{\mathsf{R}}}(\beta h(\gamma)) = x.
\end{aligned}$$

This proves the claim. ∎

6.7.3 Definition

Let $\mathcal{C}$ be a class of morphisms and let $\gamma_1, \gamma_2 \in \beta(B^*)$, for some alphabet B.

- Let R be a typed stamp over A. We say that R satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$, if it satisfies the equation with respect to each $\mathcal{C}$-morphism $h\colon B^* \to A^*$.
- Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps. We say that $\mathbf{V}$ satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$, if each typed stamp in $\mathbf{V}$ satisfies it with respect to $\mathcal{C}$.

Recall that typed stamps are actually just "discrete dense stamps" and hence, similarly to typed stamps, if $\mathsf{R} = (A, \mu, M, p, X)$ is a dense stamp and X is a compact space then ι_{R} has a unique continuous extension $\widehat{\iota_{\mathsf{R}}}\colon \beta(A^*) \to X$ which is defined by

$$L \in \widehat{\iota_{\mathsf{R}}}(\gamma) \Leftrightarrow \iota_{\mathsf{R}}^{-1}(L) \in \gamma.$$

Observe that if $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$ are dense stamps and $\Psi\colon \mathsf{R} \to \mathsf{S}$ is a morphism, then

$$\Psi \circ \widehat{\iota_{\mathsf{R}}}(\gamma) = \widehat{\iota_{\mathsf{S}}}(\gamma)$$

since for each $w \in A^*$, the equality $\Psi \circ \iota_{\mathsf{R}}(w) = \iota_{\mathsf{S}}(w)$ holds, $\iota_{\mathsf{R}}(A^*)$ is dense in X and $\Psi\colon X \to Y$ is continuous.

6.7.4 Proposition

Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps and let $\gamma_1, \gamma_2 \in \beta(A^*)$. Then the following statements are equivalent:

1. Each $\mathsf{R} \in \mathbf{V}$ satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$.
2. The dense pro-$\mathbf{V}$ stamp over A satisfies $\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) = \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2)$.

Proof. Assume that $\gamma_1, \gamma_2 \in \beta(A^*)$ such that

$$\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) = \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2)$$

holds. We prove that 1. holds by going inductively from members of $\mathcal{I}_A(\mathbf{V})$ to general members of $\mathbf{V}$.

If $\mathsf{R} \in \mathcal{I}_A(\mathbf{V})$, by definition of $\widehat{\mathsf{R}}_A(\mathbf{V})$, there exists a morphism of dense stamps $\Psi\colon \widehat{\mathsf{R}}_A(\mathbf{V}) \to \mathsf{R}$ such that

$$\widehat{\iota_{\mathsf{R}}}(\gamma_1) = \Psi \circ \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) = \Psi \circ \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2) = \widehat{\iota_{\mathsf{R}}}(\gamma_2).$$

Which proves the claim for each element of $\mathcal{I}_A(\mathbf{V})$.

If $\mathsf{R} \in \mathbf{V}(A)$, we recall that R is a trivial extension of some typed stamp $\mathsf{S} \in \mathcal{I}_A(\mathbf{V})$, where $\mathsf{R} = (A, \mu, M, p, X)$ and $\mathsf{S} = (A, \nu, N, q, Y)$. Hence there exists a morphism

$\Psi\colon \mathsf{R} \to \mathsf{S}$, such that $\Psi\colon X \to Y$ is a bijection with inverse Ψ^{-1} and since $\widehat{\iota_{\mathsf{S}}}(\gamma_1) = \widehat{\iota_{\mathsf{S}}}(\gamma_2)$, we also have

$$\widehat{\iota_{\mathsf{R}}}(\gamma_1) = \Psi^{-1} \circ \widehat{\iota_{\mathsf{S}}}(\gamma_1) = \Psi^{-1} \circ \widehat{\iota_{\mathsf{S}}}(\gamma_2) = \widehat{\iota_{\mathsf{R}}}(\gamma_2).$$

Hence the claim holds for all $\mathsf{R} \in \mathbf{V}(A)$.

If $\mathsf{R} \in \mathbf{V}$ is a typed stamp over the alphabet B, then for each $\mathcal{C}$-morphism $h\colon A^* \to B^*$, the typed stamp $\mathsf{R} \cdot h$ is in $\mathbf{V}(A)$, hence

$$\widehat{\iota_{\mathsf{R}\cdot h}}(\gamma_1) = \widehat{\iota_{\mathsf{R}\cdot h}}(\gamma_2).$$

By Lemma 6.7.2, R satisfies $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$, which proves one direction of the claim.

For the converse direction, recall that the dense pro-$\mathbf{V}$ stamp $\widehat{\mathsf{R}}_A(\mathbf{V})$ is the projective limit of the system $\mathcal{I}_A(\mathbf{V})$ and hence for any $u, v \in A^*$, it holds that $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(u) \neq \iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}(v)$ if and only if there exists a typed stamp $\mathsf{R} \in \mathcal{I}_A(\mathbf{V})$ such that $\iota_{\mathsf{R}}(u) \neq \iota_{\mathsf{R}}(v)$. Hence the same property holds for the continuous extensions, in the sense that

$$\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) \neq \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2) \Leftrightarrow \exists\, \mathsf{R} \in \mathcal{I}_A(\mathbf{V}) \;:\; \widehat{\iota_{\mathsf{R}}}(\gamma_1) \neq \widehat{\iota_{\mathsf{R}}}(\gamma_2).$$

Since 1. implies in particular that $\widehat{\iota_{\mathsf{R}}}(\gamma_1) = \widehat{\iota_{\mathsf{R}}}(\gamma_2)$ for all $\mathsf{R} \in \mathcal{I}_A(\mathbf{V})$, 2. holds. ■

6.7.5 Definition

Given a set of equations E, we denote the class of all typed stamps satisfying all equations from E with respect to $\mathcal{C}$ by $[\![E]\!]_{\mathcal{C}}$.

6.7.6 Proposition

If E is a set of equations, then $[\![E]\!]_{\mathcal{C}}$ is a $\mathcal{C}$-stream of typed stamps.

Proof. It is straight forward to see, that the intersection of $\mathcal{C}$-streams of typed stamps is a $\mathcal{C}$-stream of typed stamps. Hence it suffices to verify that for any $\gamma_1, \gamma_2 \in \beta(A^*)$, where A is some alphabet, the class $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is a $\mathcal{C}$-stream of typed stamps. Assume without loss of generality that all typed stamps in consideration are restricted.

Recall that by Lemma 6.7.2, a typed stamp over the alphabet B satisfies $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$ if and only if for all $\mathcal{C}$-morphisms $h\colon A^* \to B^*$, it satisfies $[\beta h(\gamma_1) \leftrightarrow \beta h(\gamma_2)]$. Hence we assume that the typed stamps are defined over A in the first place. The cases for arbitrary B follow completely analogously, replacing γ_1 (resp. γ_2) by $\beta h(\gamma_1)$ (resp. $\beta h(\gamma_2)$).

Suppose that $\mathsf{R} = (A, \mu, M, p, X)$ satisfies $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$. Moreover let $\mathsf{S} = (A, \nu, N, q, Y)$. We verify that $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is closed under division, trivial extension, finite direct products and $\mathcal{C}$-transformations.

Suppose S divides R. Then there exists a morphism $\Psi\colon \mathsf{R} \to \mathsf{S}$ such that $\Psi \circ \widehat{\iota_{\mathsf{R}}} = \widehat{\iota_{\mathsf{S}}}$. Hence $\widehat{\iota_{\mathsf{S}}}(\gamma_1) = \widehat{\iota_{\mathsf{S}}}(\gamma_2)$ and the class $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is closed under division.

Suppose S is a trivial extension of R. Then there exists a morphism $\Psi\colon \mathsf{S} \to \mathsf{R}$ with $\Psi\colon X \to Y$ a bijection with inverse Ψ^{-1} and thus

$$\widehat{\iota_{\mathsf{S}}}(\gamma_1) = \Psi^{-1} \circ \widehat{\iota_{\mathsf{R}}}(\gamma_1) = \Psi^{-1} \circ \widehat{\iota_{\mathsf{R}}}(\gamma_2) = \widehat{\iota_{\mathsf{S}}}(\gamma_2).$$

Thus $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is closed under trivial extension.

Suppose also S satisfies $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$. Let $\gamma \in \beta(A^*)$. Then, for each $(x, y) \in X \times Y$, it holds that

$$\widehat{\iota_{\mathsf{R}\times\mathsf{S}}}(\gamma) = (x, y) \Leftrightarrow \iota^{-1}_{\mathsf{R}\times\mathsf{S}}(x, y) \in \gamma \Leftrightarrow \iota^{-1}_{\mathsf{R}}(x) \cap \iota^{-1}_{\mathsf{S}}(y) \in \gamma.$$

Since ultrafilters are closed under supersets and intersection, we have that $\iota^{-1}_{\mathsf{R}}(x) \cap \iota^{-1}_{\mathsf{S}}(y) \in \gamma$ if and only if $\iota^{-1}_{\mathsf{R}}(x) \in \gamma$ (equiv. $\widehat{\iota_{\mathsf{R}}}(\gamma) = x$) and $\iota^{-1}_{\mathsf{S}}(y) \in \gamma$ (equiv. $\widehat{\iota_{\mathsf{S}}}(\gamma) = y$). Combining those observations, it follows that

$$\widehat{\iota_{\mathsf{R}\times\mathsf{S}}}(\gamma) = (\widehat{\iota_{\mathsf{R}}}(\gamma), \widehat{\iota_{\mathsf{S}}}(\gamma)),$$

which implies that $\mathsf{R} \times \mathsf{S}$ satisfies the equation $[\gamma_1 \leftrightarrow \gamma_2]$. Hence $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is closed under finite direct products.

Finally, let $h\colon B^* \to A^*$ be a $\mathcal{C}$-morphism. We claim that $\mathsf{T} = \mathsf{R} \cdot h$ satisfies $[\gamma_1 \leftrightarrow \gamma_2]$. Let $\varphi\colon A^* \to B^*$ be a $\mathcal{C}$-morphism, then

$$\widehat{\iota_{\mathsf{T}\cdot\varphi}} = \widehat{\iota_{\mathsf{R}\cdot(h\circ\varphi)}}.$$

Since $\mathcal{C}$-morphisms are closed under composition, $h \circ \varphi$ is a $\mathcal{C}$-morphism and since R satisfies the equation, we have $\widehat{\iota_{\mathsf{R}\cdot g}}(\gamma_1) = \widehat{\iota_{\mathsf{R}\cdot g}}(\gamma_2)$ for each $\mathcal{C}$-morphism $g\colon A^* \to A^*$, in particular for $g = h \circ \varphi$. Hence also T satisfies $[\gamma_1 \leftrightarrow \gamma_2]$ with respect to $\mathcal{C}$ and thus $[\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$ is a $\mathcal{C}$-stream. ∎

We say that a $\mathcal{C}$-stream of typed stamps $\mathbf{V}$ is defined by a set of equations E, if $\mathbf{V} = [\![\gamma_1 \leftrightarrow \gamma_2]\!]_{\mathcal{C}}$.

6.7.7 Theorem

A class of typed stamps $\mathbf{V}$ is a $\mathcal{C}$-stream of typed stamps if and only if it is defined by a set of equations with respect to $\mathcal{C}$.

Proof. By Proposition 6.7.6 if follows, that any class of typed stamps defined by a set of equations with respect to $\mathcal{C}$ forms a $\mathcal{C}$-stream of typed stamps.

For the converse direction, let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps and let $E_{\mathbf{V}}$ be the class of equations that are satisfied by each element of $\mathbf{V}$ with respect to $\mathcal{C}$. We show that these equations already define $\mathbf{V}$ as a $\mathcal{C}$-variety: Let $\mathbf{W} = [\![E_{\mathbf{V}}]\!]_{\mathcal{C}}$, then obviously $\mathbf{V} \subseteq \mathbf{W}$.

We can assume without loss of generality, that each equation in $E_{\mathbf{V}}$ is of the form $[\gamma_1 \leftrightarrow \gamma_2]$ for some $\gamma_1, \gamma_2 \in \beta(A^*)$. For if this is not that case and $\gamma_1, \gamma_2 \in \beta(B^*)$, we

replace $[\gamma_1 \leftrightarrow \gamma_2]$ by the set of equations $[\beta h(\gamma_1) \leftrightarrow \beta h(\gamma_2)]$, where $h\colon B^* \to A^*$ is a $\mathcal{C}$-morphism (this is all right, since $E_{\mathbf{V}}$ needs not to be finite).

To prove the inclusion $\mathbf{V} \supseteq \mathbf{W}$, we show that $\mathcal{I}_A(\mathbf{V}) \supseteq \mathcal{I}_A(\mathbf{W})$. Let $\mathsf{S} \in \mathcal{I}_A(\mathbf{W})$.

It follows from Proposition 6.7.4, that for any $\gamma_1, \gamma_2 \in \beta(A^*)$

$$[\gamma_1 \leftrightarrow \gamma_2] \in E_{\mathbf{V}} \text{ if and only if } \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) = \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2).$$

In particular, since $\mathsf{S} \in \mathcal{I}_A(\mathbf{W}) \subseteq [\![E_{\mathbf{V}}]\!]_{\mathcal{C}}$

$$\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_1) = \widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}(\gamma_2) \Rightarrow \widehat{\iota_{\mathsf{S}}}(\gamma_1) = \widehat{\iota_{\mathsf{S}}}(\gamma_2)$$

and thus $\widehat{\iota_{\mathsf{S}}}$ factors through $\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}$. Hence there exists a continuous function f such that $\iota_{\mathsf{S}} = f \circ \iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}$. We conclude that each language recognised by S is also a language recognised by $\widehat{\mathsf{R}}_A(\mathbf{V})$ and hence the languages recognised by S form a finite Boolean subalgebra $\mathcal{B}$ of $\mathcal{V}(A)$. Since $\mathsf{S} \in \mathcal{I}_A(\mathbf{W})$ we obtain that S is the syntactic typed stamps of $\mathcal{B}$ and since $\mathcal{B} \subseteq \mathcal{V}(A)$, we have that $\mathsf{S} \in \mathcal{I}_A(\mathbf{V})$, which proves the claim. ∎

Concrete Equations Calculated

We are now going to determine the characterising equations for the streams of typed stamps $\mathbf{V}$ from the previous section, in which we identified their dense pro-$\mathbf{V}$ stamps.

Observe that Proposition 6.7.4 states that characterising the equations that define a stream of typed stamps $\mathbf{V}$ is equivalent to characterising the kernel of $\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}}$. In fact, the equations *are* the kernel. We will thus first examine the continuous extension of $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V})}$ and then describe the kernel of the extension.

The Finite Co-Finite Boolean Algebra

Recall that the dense pro-$\mathbf{V}_{\text{Co-Fin}}$ stamp was given by

$$\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}}) = (A^*, \mathrm{id}_{A^*}, A^*, i_{A^*}, A^* \cup \{\infty\})$$

where i_{A^*} is the inclusion of A^*, which results in

$$\iota_{\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}})}(w) = \{L \in \mathcal{V}_{\text{Co-Fin}}(A) \mid w \in L\} = w.$$

Observe that since $\iota_{\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}})}$ is the canonical inclusion of A^* in the Stone space of $\mathcal{V}_{\text{Co-Fin}}(A)$, its continuous extension is the quotient map from $\beta(A^*)$ to $X_{\mathcal{V}_{\text{Co-Fin}}(A)}$ and hence for $\gamma \in \beta(A^*)$, we obtain

$$\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}})}}(\gamma) = \{L \in \mathcal{V}_{\text{Co-Fin}}(A) \mid L \in \gamma\}$$

which consequently results in

$$\widehat{\iota_{\widehat{\mathsf{R}}_A(\mathbf{V}_{\text{Co-Fin}})}}(\gamma) = \begin{cases} w & \text{if } \{w\} \in \gamma \\ \infty & \text{otherwise.} \end{cases}$$

Let $\gamma_1, \gamma_2 \in \beta(A^*)$. Regarding the kernel of $\iota_{\widehat{\mathsf{R}_A(\mathbf{V}_{\text{Co-Fin}})}}$ provides us with the defining equations of the *ne*-stream $\mathbf{V}_{\text{Co-Fin}}$, where $[\gamma_1 \leftrightarrow \gamma_2]$ is an equation for $\mathbf{V}_{\text{Co-Fin}}$, if for all $w \in A^*$

$$\{w\} \in \gamma_1 \Leftrightarrow \{w\} \in \gamma_2.$$

The Non-Regular Boolean Algebra

Recall that the Boolean algebra $\mathcal{V}_\Lambda(A)$ was generated by the finite sets $\overline{n} = \{a^n b^n \mid a, b \in A\}$ for $n \in \mathbb{N}$ and the set $\Lambda = \{a^n b^n \mid n \in \mathbb{N},\ a, b \in A\}$ and its Stone space is the set $\mathbb{N} \cup \{-1, \infty\}$, where $\mathbb{N}$ corresponds to the ultrafilters containing the set $\overline{n}$, ∞ is the ultrafilter of co-finite sets relative to Λ and -1 the ultrafilter consisting of all supersets of Λ^c. Hence, in a similar fashion as in the previous example, we obtain

$$\iota_{\widehat{\mathsf{R}_A(\mathbf{V}_\Lambda)}}(\gamma) = \begin{cases} n & \text{if } \{a^n b^n \mid a, b \in A\} \in \gamma \\ -1 & \text{if } \Lambda^c \in \gamma, \\ \infty & \text{otherwise.} \end{cases}$$

Which determines the equations in quite the same fashion as above.

6.8 Summary

Based on typed monoids, we introduced the notion of typed stamps for the recognition of (Boolean algebras of) non-regular languages. Subsequently, we defined the notions usually necessary for treating recognition of languages, such as morphisms of typed stamps (6.1.2) and the syntactic typed stamp (6.1.10), which is the smallest typed stamp recognising a finite Boolean algebra of languages. Already here it became evident that typed stamps and Stone spaces are closely related, since the last component of a typed stamp is isomorphic to the Stone space of the recognised languages.

With the final goal in mind to characterise classes of languages described by fragments of logic, which form Boolean algebras and usually are only closed under certain inverse morphisms (for instance, length preserving or non-erasing ones), we introduced $\mathcal{C}$-streams of typed stamps (6.2.2). Those proved to have an Eilenberg-like relationship (6.3.4) to $\mathcal{C}$-streams of languages.

To obtain a profinite object, as the free profinite monoid for the regular languages (or the free profinite Ext-algebra of the previous chapter), it was necessary to cut our focus down to smaller classes of typed stamps, since $\mathcal{C}$-streams of typed stamps contain also typed stamps with a monoid component so unnecessarily big that it results in the loss of all algebraic information.

Those classes (called the inherent class of a $\mathcal{C}$-stream (6.3.10)), also provide and Eilenberg-like relationship to $\mathcal{C}$-streams of languages and were shown to form a projective system and to have a projective limit in a larger category, the category of

dense stamps (6.4.3). In this category, we were able to first describe the projective limit in a rather technical manner and to then prove that with the aid of the Eilenberg-like correspondence it may also be characterised in a more accessible way: For instance, the monoid-component is actually the syntactic monoid of the associated $\mathcal{C}$-stream of languages (6.5.1) and the topological-space-component is isomorphic to the Stone space of the associated $\mathcal{C}$-stream of languages (6.5.4).

In particular, it was possible to identify projective limits of $\mathcal{C}$-streams of typed stamps as appropriate recognisers for the corresponding (possibly infinite) Boolean algebras of languages.

Using these characterisations, we calculated two small examples for two concrete $\mathcal{C}$-streams of languages.

Moreover, we were able to prove a Reiterman-like theorem for $\mathcal{C}$-streams of typed stamps, showing that each $\mathcal{C}$-stream of typed stamps is uniquely determined by a set of ultrafilter equations. These equations were subsequently calculated for the previously considered two $\mathcal{C}$-streams.

6.9 Further Research

A considerably simple yet interesting case study would be to consider equations for fragments of logic with quantifier depth one, as for instance in [GKP16]. Equations for such classes hopefully, with the contents of the following chapter, provide an inductive base for finding equations for classes of greater quantifier depth.

An obvious direction of research would be to apply these results to concrete $\mathcal{C}$-streams of languages such as $\mathbf{FO}[\mathcal{N}]$ or classes that are comparably easier to understand, such as the languages recognised by polynomial size circuits with modulo gates of depth two, and to find equations for them. In fact, the following chapter aims at building a framework for capturing classes described by fragments of logic.

Also, the framework could be extended to work with lattices instead of just Boolean algebras. Lattices are similar to Boolean algebras, but not closed under complementation. For instance, all languages of well-matched words relative to A^* form a lattice. For those, one could consider ordered typed stamps - ordered monoids were considered by Pin [Pin12] to study lattices of regular languages.

7

The Block Product: Finite and Pro-Finite

In this chapter, we use the decomposition principles of substitution and transduction, which intuitively describe the nesting of quantifiers and predicates in logical formulae and translate them to typed stamps in the following sense: For each quantifier $\mathcal{Q}$ and typed stamp R (which has a certain relation to that quantifier) and for each predicate and typed stamp S (which in turn is in some relation to the predicate), we may build a new typed stamp from R and S that recognises the same languages as the formula binding a variable in the predicate by the quantifier.

In particular, we view these principles in the light of duality. This enables us to formulate said principle also for dense stamps (in particular projective limits of typed stamps) and give a relation to logic, which exceeds the necessity of finiteness (in the sense the we may now consider not only one predicate and one quantifier, but also infinitely many).

7.1 Typed Stamps and Decomposition in Logic

Before drawing a connection between typed stamps and languages described by logical formulae, we need to make some assumptions to avoid notational clutter. The main reason for that is that the concepts we are treating to understand logical formulae work on formulae with multiple free variables, but are applied one variable at a time, which makes it extremely cumbersome to always specify the current set of free variables and the distinguished free variable in consideration, especially when applied multiple times. We thus use the following simplification:

Let $\mathcal{X}$ be a set of free variables and let x be a distinguished free variable, not contained in $\mathcal{X}$. Moreover let ϕ be a formula over the alphabet A whose free variables are contained in $\mathcal{X} \cup \{x\}$. If x occurs free in ϕ, then we treat ϕ as a formula with precisely

one free variable x and assume the rest of the variables in $\mathcal{X}$ to be encoded in the alphabet in the sense that ϕ defines a $\{x\}$-structure language over $(A \times 2^{\mathcal{X}})^* \otimes \{x\}$. Otherwise, that is if ϕ does not contain x as a free variable, then we treat ϕ as a sentence over $(A \times 2^{\mathcal{X}})^*$.

The benefit of this lies in the fact that we can now, without loss of generality, consider formulae with at most one free variable, which prevents us from having to specify the variable under consideration and the set of remaining free variables. Remarks are provided at appropriate points, to illustrate how this principle is being used.

Notation

Let R be a typed stamp over the marked alphabet $(A \times 2^{\{x\}})^*$. We say that R recognises a language $L_x \subseteq A^* \otimes \{x\}$ if $L_x = S \cap (A^* \otimes \{x\})$ for some $S \in L(\mathsf{R})$ and that R recognises a language $L \subseteq A^*$, if $\mathsf{R} \cdot i_{A^*}$ recognises L, where $i_{A^*} \colon A^* \to (A \times 2^{\{x\}})^*$ is the morphism sending a to $(a, \emptyset)$.

Moreover, we write $L_{[x]}(\mathsf{R})$ for the set of languages over $A^* \otimes \{x\}$ recognised by R and $L_{[\emptyset]}(\mathsf{R})$ for the set of languages over A^* recognised by R. By $X_{\mathsf{R}[x]}$ we denote the set $\iota_{\mathsf{R}}(A^* \otimes \{x\})$.

Let $\mathcal{S}$ be a Boolean algebra over the alphabet A containing formulae with at most one free variable x. Then we denote by $\mathcal{S}[x]$ the Boolean subalgebra of $\mathcal{S}$ containing only formulae in which x occurs free and by $\mathcal{S}[\emptyset]$ the Boolean subalgebra of sentences of $\mathcal{S}$.

We say that R recognises $L(\mathcal{S})$, if it recognises each language in $L(\mathcal{S}[x])$ and each language in $L(\mathcal{S}[\emptyset])$.

Transductions and Substitutions

Decomposition of logical formulae is a natural way to understand the evaluation of a sentence in some logical formalism on a word. For instance, let ϕ be a formula with a free variable x and say we are interested whether the word $w = w_1 \dots w_n$ satisfies the sentence Maj x $\phi(x)$. The intuitive approach to solve this issue is to evaluate for each position $i = 1, \dots, n$, whether $w_{x=i}$ models $\phi(x)$, keep those values in mind and then check whether the majority of them was true. In this case, the sentence is decomposed into quantifier and subformula.

The concepts of substitution and transduction, which we will be treating in the following, are precisely the formalisations of the procedure we just described, to understand logical formulae on words. In order to make the formal definitions more accessible, we illustrate them on one further example.

As before, we consider the formula Maj x $\phi(x)$ and let $w = w_1 \dots w_n \in A^*$. The transduction induced by $\phi(x)$ is a map $\tau_\phi \colon A^* \to (\{\phi, \neg\phi\})^*$ such that $\tau_\phi(w)$ is the

word $v_1 \dots v_n$ with

$$v_i = \begin{cases} \phi & \text{if } w_{x=i} \models \phi(x) \\ \neg\phi & \text{otherwise.} \end{cases}$$

Applying the transduction amounts to the step of evaluating for each $i = 1, \dots, n$ whether $w_{x=i}$ models $\phi(x)$ and keeping the information encoded in the new alphabet. In particular

$$w \models \text{Maj } x\ \phi(x) \Leftrightarrow \tau_\phi(w) \models \text{Maj } x\ Q_\phi(x).$$

This equivalence describes precisely that after the previous evaluation of the single values for each position, it is no longer necessary to consider $\phi(x)$, since we only need to check whether the majority of the values now encoded in the alphabet, is true.

Replacing the formula Maj $x\ Q_\phi(x)$ by Maj $x\ \phi(x)$ is the counterpart to the transduction induced by ϕ and called the substitution σ_ϕ induced by ϕ.

All together, we obtain

$$\tau_\phi(w) \models \text{Maj } x\ Q_\phi(x) \Leftrightarrow w \models \sigma_\phi(\text{Maj } x\ Q_\phi(x)).$$

Transduction and Substitutions: Local Versions

In the following, we assume that all formulae are part of some common fragment of logic $\mathcal{Q}[\mathcal{N}]$.

Substitution

We now formally define substitution and transduction for Boolean algebras of formulae and subsequently consider their connections to typed stamps.

7.1.1 Definition

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x. Moreover, let $\mathcal{R}$ be a finite Boolean algebra of sentences over the alphabet $X_{\mathcal{S}[x]}$. A substitution is a map

$$\sigma_{\mathcal{S}} \colon \mathcal{R} \to \mathcal{Q}_A[\mathcal{N}],$$

that sends any formula in $\phi \in \mathcal{R}$ to the formula obtained, when replacing each occurrence of $Q_\psi(y)$ in ϕ by $\psi(x := y)$, which is the formula ψ when renaming each occurrence of x in ψ to y.

Observe that by renaming variables, we may assume that the variables in $\mathcal{S}$ and $\mathcal{R}$ are distinct, and in particular that x does not appear as a bound variable in any formula of $\mathcal{R}$.

We let

$$\mathcal{R} \circ \mathcal{S} = \sigma_{\mathcal{S}}(\mathcal{R}) \cup \mathcal{S}[\emptyset].$$

For instance, if $\mathcal{S} = \{\top, \phi, \neg\phi, \bot\}$, where x occurs free in ϕ, then Maj y $Q_\phi(y)$ is mapped by $\sigma_\mathcal{S}$ to Maj y $\phi(y)$.

Illustration on multiple free variables: As indicated before, we are able to use substitution on formulae with multiple free variables, but do not state it explicitly due to technical reasons. For instance, let $\phi(x, y) = x < y$. Then the substitution mapping $\exists\ z\ Q_{x<y}(z)$ to $\exists\ z\ z < y$, needs to specify that x is being substituted, not y. By considering $\phi(x, y) = x < y$ as a formula with one free variable over the alphabet $(A \times 2^{\{y\}})^*$, this problem is avoided. In particular, if the formulae in $\mathcal{S}$ had free variables in some set $\mathcal{X} \cup \{x\}$, then the formulae in $\mathcal{R} \circ \mathcal{S}$ have free variables in $\mathcal{X}$ and define languages over the common alphabet $(A \times 2^\mathcal{X})$.

7.1.2 Proposition

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x and let $\mathcal{R}$ be a finite Boolean algebra of sentences over the alphabet $X_{\mathcal{S}[x]}$. Then

$$\sigma_\mathcal{S} \colon \mathcal{R} \to \mathcal{R} \circ \mathcal{S}$$

is a morphism of Boolean algebras.

Proof sketch: It suffices to observe that by definition, substitution distributes over Boolean connectives. The rest of the proof is a straight-forward induction over the structure of formulae and thus omitted.

Transductions

As a counterpart to substitutions, we define the transduction induced by an appropriate Boolean algebra of formulae $\mathcal{S}$:

7.1.3 Definition

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x. The transduction induced by $\mathcal{S}$ is the map $\tau_\mathcal{S} \colon A^* \to X^*_{\mathcal{S}[x]}$ defined by

$$\tau_\mathcal{S}(w) = \kappa_\mathcal{S}(w_{x=1}) \cdots \kappa_\mathcal{S}(w_{x=|w|})$$

where $\kappa_\mathcal{S} \colon A^* \otimes \{x\} \to X_{\mathcal{S}[x]}$ is the map sending $w_{x=i}$ to the unique element $\phi \in X_{\mathcal{S}[x]}$ for which $w_{x=i} \models \phi$.

Observe that the map $\kappa_\mathcal{S}$ is precisely the dual map to the inclusion

$$\begin{aligned} i_\mathcal{S} \colon \mathcal{S} &\hookrightarrow \mathcal{P}(A^* \otimes \{x\}) \\ \phi &\mapsto L_\phi, \end{aligned}$$

since $i_\mathcal{S}(\phi) = \{w_{x=i} \in A^* \otimes \{x\} \mid w_{x=i} \models \phi\} = \kappa_\mathcal{S}^{-1}(\phi)$.

7.1.4 Proposition

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x. Moreover, let $\mathcal{R}$ be a finite Boolean algebra of sentences over the alphabet $X_{\mathcal{S}[x]}$. Then for each $\phi \in \mathcal{R}$ and $w \in A^*$ the equivalence

$$w \in L_{\sigma_{\mathcal{S}}(\phi)} \Leftrightarrow \tau_{\mathcal{S}}(w) \in L_\phi$$

holds.

Proof sketch: Let $\psi \in X_{\mathcal{S}[x]}$. Observe that $\tau_{\mathcal{S}}(w)_{x=i} \models Q_\psi(x)$ if and only if $(\tau_{\mathcal{S}}(w))_i = \psi$, which, by definition of $\tau_{\mathcal{S}}$ is the case if and only if $w_{x=i} \models \psi$. Since $\sigma_{\mathcal{S}}$ replaces each occurrence of $Q_\psi(x)$ in ϕ by $\psi(x)$ and a quantifier depends only on the truth values for each position, a simple induction over the structure of formulae proves the claim.

Remark: Transductions and substitutions are related very closely by duality. Since $\sigma_{\mathcal{S}}$ is a morphism from $\mathcal{R}$ to $\mathcal{R} \circ \mathcal{S}$, it induces a morphism of Boolean algebras of languages from $L(\mathcal{R})$ to $L(\mathcal{R} \circ \mathcal{S})$ by sending L_ϕ, to $L_{\sigma_{\mathcal{S}}(\phi)}$. This morphism has a dual map $\sigma_{\mathcal{S}}^{-1} \colon X_{L(\mathcal{R}\circ\mathcal{S})} \to X_{L(\mathcal{R})}$. Since $\mathcal{R}$ consists of formulae over the alphabet $X_{\mathcal{S}[x]}$, the set of languages $L(\mathcal{R})$ forms subalgebra of $\mathcal{P}(X^*_{\mathcal{S}[x]})$ and since $\mathcal{S}$ is consists of formulae over A^*, $L(\mathcal{R} \circ \mathcal{S})$ is a subalgebra of $\mathcal{P}(A^*)$. Hence there exist maps

$$\iota_{X^*_{\mathcal{S}[x]}} \colon X^*_{\mathcal{S}[x]} \to X_{L(\mathcal{R})} \text{ and } \iota_{A^*} \colon A^* \to X_{L(\mathcal{R}\circ\mathcal{S})}$$

with dense image. And the diagram

$$\begin{array}{ccc} A^* & \xrightarrow{\tau_{\mathcal{S}}} & X^*_{\mathcal{S}[x]} \\ \downarrow{\scriptstyle \iota_{A^*}} & & \downarrow{\scriptstyle \iota_{X^*_{\mathcal{S}[x]}}} \\ X_{L(\mathcal{R}\circ\mathcal{S})} & \xrightarrow{\sigma_{\mathcal{S}}^{-1}} & X_{L(\mathcal{R})} \end{array}$$

commutes. As such, the transduction is the restriction to A^* of the dual of the substitution $\sigma_{\mathcal{S}} \colon L(\mathcal{R}) \to L(\mathcal{R} \circ \mathcal{S})$.

As a consequence of Proposition 7.1.4, we may characterise the languages in $L(\mathcal{R} \circ \mathcal{S})$.

7.1.5 Corollary

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x and let $\mathcal{R}$ be a finite Boolean algebra of sentences over the alphabet $X_{\mathcal{S}[x]}$.
Then, a language belongs to $L(\mathcal{R} \circ \mathcal{S})$ if and only if it is a Boolean combination of languages in $L(\mathcal{S}[\emptyset])$ and languages of the form $\tau_{\mathcal{S}}^{-1}(L)$, where $L \in L(\mathcal{R})$.

Up to now, transductions and substitutions have been defined purely on the logic side. We now translate these principles to languages and successively to typed stamps.

7.1.6 Definition

Let $\mathcal{L}$ be a finite Boolean algebra of languages over $A^* \otimes \{x\}$. The transduction induced by $\mathcal{L}$ is the map $\tau_{\mathcal{L}} \colon A^* \to X_{\mathcal{L}}^*$ defined by

$$\tau_{\mathcal{L}}(w) = \kappa_{\mathcal{L}}(w_{x=1}) \cdot \dots \cdot \kappa_{\mathcal{L}}(w_{x=|w|})$$

where $\kappa_{\mathcal{L}} \colon A^* \otimes \{x\} \to X_{\mathcal{L}}$ is the map sending $w_{x=i}$ to the unique element $L \in X_{\mathcal{L}}$ for which $w_{x=i} \in L$.

If $\mathcal{S}$ is a finite Boolean algebra of formulae with at most one free variable x, then $X_{\mathcal{S}[x]}$ and $X_{L(\mathcal{S}[x])}$ are isomorphic as sets and $\kappa_{L(\mathcal{S}[x])} = \kappa_{\mathcal{S}[x]}$ modulo that isomorphism, which in turn provides that

$$\tau_{L(\mathcal{S})} = \tau_{\mathcal{S}}.$$

Similarly, any typed stamp over a marked alphabet induces a transduction in the following way:

7.1.7 Definition

Let S be a typed stamp over $A^* \times 2^{\{x\}}$. The transduction induced by S is the map $\tau_{\mathsf{S}} \colon A^* \to X_{\mathsf{S}[x]}^*$ defined by

$$\tau_{\mathsf{S}}(w) = \iota_{\mathsf{S}}(w_{x=1}) \dots \iota_{\mathsf{S}}(w_{x=|w|}).$$

In a similar fashion as before, $X_{\mathsf{S}[x]}$ and the Stone space of $L_{[x]}(\mathsf{S})$ are isomorphic and thus

$$\tau_{\mathsf{S}} = \tau_{L_{[x]}(\mathsf{S})}.$$

Summarising those observations, we obtain

7.1.8 Proposition

Let S be a typed stamp over the alphabet $(A \times 2^{\{x\}})^*$ and $\mathcal{S}$ a Boolean algebra of formulae with at most one free variable such that $L(\mathsf{S}[x]) = L(\mathcal{S}[x])$. Then $X_{L(\mathcal{S}[x])}$ and $X_{L_{[x]}(\mathsf{S})}$ are isomorphic as sets and modulo isomorphism

$$\tau_{\mathcal{S}} = \tau_{L(\mathcal{S}[x])} = \tau_{\mathsf{S}}.$$

The proof of the three equalities is straight forward together with the rough sketches given previously.

7.1.9 Corollary

Let S be a typed stamp over the alphabet $(A \times 2^{\{x\}})^*$ and $\mathcal{S}$ a Boolean algebra of formulae with at most one free variable x such that $L(\mathsf{S}[x]) = L(\mathcal{S}[x])$. Moreover let R be a typed stamp over $X_{\mathsf{S}[x]}$ and $\mathcal{R}$ a Boolean algebra of sentences over the same alphabet such that $L(\mathsf{R}) = L(\mathcal{R})$. Then $L(\mathcal{R} \circ \mathcal{S})$ is the Boolean algebra generated by all sets $\tau_{\mathsf{S}}^{-1}(L)$ where $L \in L(\mathsf{R})$ and the sets $L_{[\emptyset]}(\mathsf{S})$.

Hence we may mirror the nesting of quantifiers on logical formulae through transductions on typed stamps. While substitution and transduction are dual to each other, we are now going to concern ourselves with a direct translation of substitution to typed stamps.

The Block Product

What substitution is to formulae, the block product is to typed stamps: We show that given any two Boolean algebras of adequate formulae $\mathcal{S}, \mathcal{R}$ such that $\mathcal{R} \circ \mathcal{S}$ is defined, then if R is a typed stamp recognising $L(\mathcal{R})$ and S a typed stamp recognising $L(\mathcal{S})$, the block product $\mathsf{R}\Box\mathsf{S}$ is a typed stamp that recognises $L(\mathcal{R} \circ \mathcal{S})$.

Let $\mathsf{R} = (X_{\mathsf{S}[x]}, \mu, M, p, X)$ and $\mathsf{S} = (A \times 2^{\{x\}}, \nu, N, q, Y)$ be typed stamps. For each $a \in A$, define the function

$$\begin{aligned} f_a \colon N \times N &\to M \\ (n_1, n_2) &\mapsto \mu(q(n_1\nu(a, \{x\})n_2)). \end{aligned}$$

Observe that this is well-defined since $X_{\mathsf{S}[x]} \subseteq Y$.

7.1.10 Definition

The block product of $\mathsf{R} = (X_{\mathsf{S}[x]}, \mu, M, p, X)$ and $\mathsf{S} = (A \times 2^{\{x\}}, \nu, N, q, Y)$ is the typed stamp

$$\mathsf{R}\Box\mathsf{S} := (A, \mu\Box\nu, M\Box N, p\Box q, X \times Y)$$

where

- $M\Box N$ is the usual block product of monoids ,
- $\mu\Box\nu$ is the morphism sending a to $(f_a, \nu(a))$, and
- $p\Box q$ sends the tuple (f, n) to $((p \circ f)(1, 1), q(n))$.

Remark: To the reader familiar with the block product for finite monoids, restricting the block product to typed stamps with matching alphabets might seem rather strong. The main reason for that restriction is that typed stamps force us to state the morphism. Thus, what is possible for finite monoids, namely to pick the "correct" morphism for the recognition of languages recognised by formulae, is

not immediately possible for typed stamps. In particular, the appropriate morphism here is part of the definition.

In the following, we assume that S is always a typed stamp over a marked alphabet and R is over the alphabet $X_{\mathsf{S}[x]}$.

7.1.11 Proposition

Let R and S be typed stamps. The languages recognised by $\mathsf{R}\Box\mathsf{S}$ are precisely the languages over A^* that are Boolean combinations of languages of the form $\tau_{\mathsf{S}}^{-1}(L)$, where $L \subseteq X_{\mathsf{S}[x]}$ is recognised by R and languages in $L_{[\emptyset]}(\mathsf{S})$.

Proof. Let $\mathsf{R} = (X_{\mathsf{S}[x]}, \mu, M, p, X)$ and let $\mathsf{S} = ((A \times 2^{\{x\}})^*, \nu, N, q, Y)$. As usual, we denote by $i_A \colon A^* \to (A \times 2^{\{x\}})^*$ the inclusion. Then the languages $L_{[\emptyset]}(\mathsf{S})$ are precisely the languages recognised by $\mathsf{S} \cdot i_A$.

Let L be a language recognised by $\mathsf{R}\Box\mathsf{S}$, such that

$$L = \iota_{\mathsf{R}\Box\mathsf{S}}^{-1}(\{(t_X, t_Y)\})$$

for some $t_X \in X$ and $t_Y \in Y$. Since each language recognised by $\mathsf{R}\Box\mathsf{S}$ is a Boolean combination of such languages, we can make this assumption without loss of generality. We define the languages

$$L_X = \iota_{\mathsf{R}}^{-1}(t_X) \text{ and } L_Y = \iota_{\mathsf{S}\cdot i_A}^{-1}(t_Y)$$

and claim that $L = \tau_{\mathsf{S}}^{-1}(L_X) \cap L_Y$. Let $w \in A^*$ and $\mu\Box\nu(w) = (f_w, \nu(w))$. By definition of the block product

$$\iota_{\mathsf{R}\Box\mathsf{S}}(w) = (t_X, t_Y) \Leftrightarrow p \circ f_w(1,1) = t_X \text{ and } \iota_{\mathsf{S}}(w) = t_Y. \tag{7.1}$$

By definition of f_w, we obtain

$$f_w(1,1) = f_{w_1}(1, \nu(w_{>1})) + \cdots + f_{w_i}(\nu(w_{<i}), \nu(w_{>i})) + \cdots + f_{w_{|w|}}(\nu(w_{<|w|}), 1)$$

where $f_{w_i}(\nu(w_{<i}), \nu(w_{>i})) = \mu(q \circ \nu(w_{<i}(w_i, x)w_{>i})) = \mu(\iota_{\mathsf{S}}(w_{x=i}))$. In particular, since μ is a morphism

$$f_w(1,1) = \mu(\iota_{\mathsf{S}}(w_{x=1}) \dots \iota_{\mathsf{S}}(w_{x=i}) \dots \iota_{\mathsf{S}}(w_{x=|w|})) = \mu \circ \tau_{\mathsf{S}}(w)$$

and hence

$$p \circ f_w(1,1) = p \circ \mu \circ \tau_{\mathsf{S}}(w) = \iota_{\mathsf{R}} \circ \tau_{\mathsf{S}}(w). \tag{7.2}$$

Since $w \in A^*$, $\iota_{\mathsf{S}}(w) = \iota_{\mathsf{S}\cdot i_A}(w)$ and by Equation 7.1

$$\begin{aligned} w \in L &\Leftrightarrow \iota_{\mathsf{R}} \circ \tau_{\mathsf{S}}(w) = t_X \text{ and } \iota_{\mathsf{S}}(w) = t_Y \\ &\Leftrightarrow \tau_{\mathsf{S}}(w) \in L_X \text{ and } w \in L_Y \\ &\Leftrightarrow w \in \tau_{\mathsf{S}}^{-1}(L_X) \cap L_Y, \end{aligned}$$

which proves one direction of the claim.

The converse direction is completely analogous. Let $L_X = \iota_{\mathsf{R}}^{-1}(\{t_X\})$ be a language recognised by R and $L_Y = \iota_{\mathsf{S}\cdot i_A}^{-1}(\{t_Y\})$ a language in $L_{[\emptyset]}(\mathsf{S})$, where $t_X \in X$ and $t_Y \in Y$. As before, it is without loss of generality possible to assume that both languages are of that form.

By Equations 7.1 and 7.2, we obtain

$$\begin{aligned} w \in \iota_{\mathsf{R}\square\mathsf{S}}^{-1}(\{(t_X, t_Y)\}) &\Leftrightarrow \iota_{\mathsf{R}} \circ \tau_{\mathsf{S}}(w) = t_X \text{ and } \iota_{\mathsf{S}}(w) = t_Y \\ &\Leftrightarrow \tau_{\mathsf{S}}(w) \in L_X \text{ and } w \in L_Y \\ &\Leftrightarrow w \in \tau_{\mathsf{S}}^{-1}(L_X) \cap L_Y, \end{aligned}$$

and hence that $\mathsf{R}\square\mathsf{S}$ recognises all Boolean combinations of such languages. ∎

7.1.12 Corollary Block Product Principle

Let $\mathcal{S}$ be a finite Boolean algebra of formulae over A with at most one free variable x and let S be a typed stamp over $A \times 2^{\{x\}}$ such that $L(\mathsf{S}) = L(\mathcal{S})$. Moreover, let $\mathcal{R}$ be a finite Boolean algebra of sentences over the alphabet $X_{\mathcal{S}[x]}$ and let R be a typed stamp over $X_{\mathsf{S}[x]}$ with $L(\mathsf{R}) = L(\mathcal{R})$. Then

$$L(\mathcal{R} \circ \mathcal{S}) = L(\mathsf{R}\square\mathsf{S}).$$

This is a direct consequence of Corollary 7.1.9 and the previous Proposition.

We conclude this section with an example, in which we build a typed stamp recognising the language defined by the formula

$$\text{Maj } y \text{ Maj } x \; x < y$$

over the alphabet A, using the block product principle from Corollary 7.1.12. While the typed stamp we will be constructing is significantly larger then the syntactic typed stamp of the language, it illustrates the basic concept: By constructing a recogniser for the innermost formulae and applying the block product with an appropriate typed stamp to simulate quantifier application, it is in general possible to inductively build typed stamps recognising languages defined by large formulae.

By abuse of notation, in this example we use a as a placeholder to represent that it may stand for any letter $a \in A$. To recognise the $\{x, y\}$-structure language defined by the predicate $<$, we define the typed stamp

$$\mathsf{R}_< := ((A \times 2^{\{y\}}) \times 2^{\{x\}}, \mu_<, M_<, \mathrm{id}_{M_<}, M_<).$$

Observe that the alphabet is not written as $(A \times 2^{\{x,y\}})$, since in the next step, we consider only x for substitution and view y not as a free variable, but as part of the alphabet. The monoid $M_<$ is given by the multiplication table on the left and $\mu_<$ is the morphism defined by the map on the right

$\cdot$	1	x	y	$+$	$-$
1	1	x	y	$+$	$-$
x	x	$-$	$+$	$-$	$-$
y	y	$-$	$-$	$-$	$-$
$+$	$+$	$-$	$-$	$-$	$-$
$-$	$-$	$-$	$-$	$-$	$-$

$$\begin{array}{ll} ((A \times 2^{\{y\}}) \times 2^{\{x\}}) & \to M_< \\ ((a, \emptyset), \{x\}) & \mapsto x \\ ((a, \{y\}), \emptyset) & \mapsto y \\ ((a, \{y\}), \{x\}) & \mapsto - \\ ((a, \emptyset), \emptyset) & \mapsto 1 \end{array}$$

The $\{x, y\}$-structure language defined by $<$ is precisely the set $\{w_{x=i,y=j} \mid i < j\}$, which is equal to $\iota^{-1}_{\mathsf{R}_<}(\{+\})$. We now use substitution to apply one Maj-quantifier. As indicated before, we encode the variables not considered for substitution – in this case y – in the alphabet. Hence the typed stamp, which we use in the block product, is defined over the alphabet $X_{\mathsf{R}_<[x]} = \iota_{\mathsf{R}_<}((A \times 2^{\{y\}})^* \otimes \{x\}) = \{+, -, x\}$.

Let

$$\mathsf{R}_{\mathrm{Maj}} := (\{+, -, x\}, \mu_{\mathrm{Maj}}, \mathbb{Z}, p_{>0}, \{+, -\}),$$

where the morphism μ_{Maj} sends $+$ to 1 and $-$ to -1 and x to 0. Moreover $p_{>0}$ maps any number larger than 0 to $+$ and otherwise to $-$. Since $\mathsf{R}_{\mathrm{Maj}}$ is defined over an appropriate alphabet, the block product of $\mathsf{R}_{\mathrm{Maj}}$ and $\mathsf{R}_<$ exists and we have

$$\mathsf{R}_{\mathrm{Maj}} \square \mathsf{R}_< = (A \times 2^{\{y\}}, \mu_{\mathrm{Maj}} \square \mu_<, \mathbb{Z} \square M_<, p_{>0} \square \mathrm{id}_{M_<}, \{+, -\} \times M_<).$$

Since the morphism $\mu_{\mathrm{Maj}} \square \mu_<$ maps a $\{y\}$-structure $w_{y=j}$ to a tuple, where the first component is a function from $M_< \times M_< \to \mathbb{Z}$ whose value on $(1, 1)$ determines whether $\iota_{\mathsf{R}_{\mathrm{Maj}} \square \mathsf{R}_<}$ maps $w_{y=j}$ to $+$ or $-$ in the first component, we consider this function, denoted by $f_{w_{y=j}}$. According to the multiplication on the block product, we have

$$\begin{aligned} f_{w_{y=j}}(1,1) = f_{(w_1,\emptyset)}(1,y) + \ldots \\ \cdots + f_{(w_{j-1},\emptyset)}(1,y) + f_{(w_j,\{y\})}(1,1) + f_{(w_{j+1},\emptyset)}(y,1) + \ldots \\ \cdots + f_{(w_{|w|},\emptyset)}(y,1). \end{aligned}$$

Recall that the values of the functions $f_{w_i}(l, r)$, where $w_i \in A \times 2^{\{y\}}$ and $l, r \in M_<$, are precisely the values $\mu_{\mathrm{Maj}}(l \cdot \mu_<(w_i, \{x\}) \cdot r)$ and we thus obtain that

$$\begin{aligned} f_{(a,\emptyset)}(1,y) &= \mu_{\mathrm{Maj}}(1 \cdot x \cdot y) = \mu_{\mathrm{Maj}}(+) = 1 \\ f_{(a,\emptyset)}(y,1) &= -1 \\ f_{(a,\{y\})}(1,1) &= \mu_{\mathrm{Maj}}(1 \cdot - \cdot 1) = -1 \end{aligned}$$

Since $\pi_1(\iota_{\mathsf{R}_{\mathrm{Maj}} \square \mathsf{R}_<}(w_{y=j})) = p_{>0} \circ f_{w_{y=j}}(1, 1) = +$ if and only if $f_{w_{y=i}}(1, 1)$ is greater than 0, inserting the values above in the equation for $f_{w_{y=j}}(1, 1)$ proves that $\pi_1(\iota_{\mathsf{R}_{\mathrm{Maj}} \square \mathsf{R}_<}(w_{y=j})) = +$ if and only if $j < \frac{|w|}{2}$. Which results in

$$A^* \otimes \{y\} \cap \iota^{-1}_{\mathsf{R}_{\mathrm{Maj}} \square \mathsf{R}_<}(\{+\} \times M_<) = \left\{ w_{y=j} \mid j < \frac{|w|}{2} \right\}$$

and shows that $\mathsf{R}_{\mathrm{Maj}}\square\mathsf{R}_<$ recognises the $\{y\}$-structure language defined by Maj x x $<$ y.

It now remains to use the block product principle to obtain a recogniser for the language defined by Maj y Maj x $x < y$. Due to the technicality of the matter and the similarity to the previous step, we omit the details of the third step and provide only a rough intuition. To apply the block product once more, the typed stamp which is being added needs to be defined over the alphabet $X_{(\mathsf{R}_{\mathrm{Maj}}\square\mathsf{R}_<)[y]} = \iota_{\mathsf{R}_{\mathrm{Maj}}\square\mathsf{R}_<}(A^* \otimes \{y\}) = \{+,-\} \times \{y\}$. Since the right side containing y provides no additional information, we identify the alphabet with $\{+,-\}$ and consider $\mathsf{R}_{\mathrm{Maj}}\square(\mathsf{R}_{\mathrm{Maj}}\square\mathsf{R}_<)$. Again, the function f_w simulates for all positions at which y may occur, whether it belongs to $\{w_{y=j} \mid j < \frac{|w|}{2}\}$, which is possible since the right hand since of the block product recognises that language, and outputs 1 if it does and -1 if it does not. It follows in a similar fashion, as the previous step, that

$$L(\text{Maj } y \text{ Maj } x \; x < y) = \iota^{-1}_{\mathsf{R}_{\mathrm{Maj}}\square(\mathsf{R}_{\mathrm{Maj}}\square\mathsf{R}_<)}(\{+\} \times (\{+,-\} \times M_<))$$

7.2 Substitution and Transduction on Streams

In the last section, we introduced substitutions, transductions and the block product, all of them for finite objects, such as finite Boolean algebras of formulae or typed stamps, which are finite also in the sense that they recognise only finite Boolean algebras of languages. We now extend these principles to possibly infinite classes of logic formulae and the most general $\mathcal{C}$-streams of typed stamps: lp-streams of typed stamps. We thus restrict ourselves to classes of formulae that define an lp-stream of languages, which in turn has a corresponding lp-stream of typed stamps.

We introduce the notion of *logic class*. Intuitively, one can think of a logic class as any common fragment of logic, such as **FO**, **MOD** or **Maj** with any set of predicates, but also subclasses of these, such as **FO**[=] formulae of depth k.

Let A and B be finite alphabets. Then, each map $r\colon A \to B$ induces a map $\rho_r\colon Q_B[\mathcal{N}] \to Q_A[\mathcal{N}]$ by replacing any occurrence of the letter predicate $Q_b(x)$ for $b \in B$ by

$$\bigvee_{r(a)=b} Q_a(x).$$

We call this map a letter substitution and it is not hard to verify, that if $\mathcal{B}$ is a Boolean algebra of formulae over B, then $\rho_r(\mathcal{B})$ is a Boolean algebra of formulae over A. Denote by $r^*\colon A^* \to B^*$ the extension of r to a morphism, then

$$r^*(w)_{x=i} \models Q_b(x) \Leftrightarrow w_{x=i} \models \bigvee_{r(a)=b} Q_a(x)$$

and for any formula ϕ over B, it follows that $w \in L_{\rho_r(\phi)} \Leftrightarrow w \in r^{*-1}(L_\phi)$.

7.2.1 Definition

A *logic class* Γ is a map that associates to each finite alphabet A a set of formulae Γ_A satisfying the following properties:

1. For each finite alphabet A, the set Γ_A is a Boolean algebra.
2. For all finite alphabets A and B, Γ is closed under letter substitutions, in the sense that if $\phi \in \Gamma_B$ and $r\colon A \to B$ is a map, then $\rho_r(\phi) \in \Gamma_A$.

We define the set of languages described by Γ over the alphabet A by

$$L_A(\Gamma) = \{L_\phi \mid \phi \in \Gamma_A \text{ and } \phi \text{ is a sentence.}\}.$$

By $L(\Gamma)$ we denote the class associating to each alphabet A, the Boolean algebra of languages $L_A(\Gamma)$. Observe that if $r\colon A \to B$ is a map between finite alphabets, it induces an lp-morphism r^* and by the previous observation that $L_{\rho_r(\phi)} = {r^*}^{-1}(L_\phi)$, the closure under letter substitutions of Γ ensures that $L(\Gamma)$ is an lp-stream of languages.

Remark: As before, we use the twist of encoding all but one free variable in the alphabet. We thus only consider logic classes in which for each alphabet, Γ_A consists only of formulae with at most one free variable. Observe that letter substitution does in this case not affect the encoding of the variables, but only the part concerned with the "actual" alphabet.

By "Γ is a logic class of sentences", we mean that for each alphabet A, Γ_A consists only of sentences and by "Λ is a logic class with at most one free variable", that each formula in Λ_A has at most one free variable.

7.2.2 Definition Subsitution Product

Let Γ be a logic class of sentences and Λ a logic class with at most one free variable x. Define the class $\Gamma \circ \Lambda$, that associates to each finite alphabet A the Boolean algebra generated by all formulae $\phi \in \mathcal{R} \circ \mathcal{S}$, where $\mathcal{S}$ is a finite subalgebra of Λ_A and $\mathcal{R}$ is a finite subalgebra of $\Gamma_{X_{\mathcal{S}[x]}}$.

The following Lemma is needed to show that $\Gamma \circ \Lambda$ is closed under letter substitution.

7.2.3 Lemma

Let Γ be a logic class of sentences, Λ a logic class with at most one free variable x and let A and B be finite alphabets, where $r\colon A \to B$ is a map. Moreover let $\mathcal{S}$ be a finite subalgebra of Λ_B and $\mathcal{R}$ a finite subalgebra of $\Gamma_{X_{\mathcal{S}[x]}}$. Then there exists a map $s\colon X_{\rho_r(\mathcal{S})[x]} \to X_{\mathcal{S}[x]}$ such that

$$\rho_r \circ \sigma_{\mathcal{S}}(\psi) = \sigma_{\rho_r(\mathcal{S})} \circ \rho_s(\psi)$$

for all $\psi \in \mathcal{R}$.

Proof. It is clear from the definition of ρ_r, that it induces a Boolean algebra morphism from $\mathcal{S}$ to its image $\rho_r(\mathcal{S})$. In particular, $\rho_r(\mathcal{S}[x]) = \rho_r(\mathcal{S})[x]$. As a morphism of Boolean algebras, ρ_r has a dual, denoted by $\rho_r^{-1}\colon X_{\rho_r(\mathcal{S})} \to X_{\mathcal{S}}$. We let $s\colon X_{\rho_r(\mathcal{S})[x]} \to X_{\mathcal{S}[x]}$ be the restriction of ρ_r^{-1} to $X_{\rho_r(\mathcal{S})[x]}$.

Recall that for any two formulae $\phi \in X_{\rho_r(\mathcal{S})}$ and $\psi \in X_{\mathcal{S}}$ the dual ρ_r^{-1} is defined by

$$\rho_r^{-1}(\phi) = \psi \Leftrightarrow \phi = \rho_r(\psi).$$

Each formula in $\mathcal{R}$ is defined over the alphabet $X_{\mathcal{S}[x]}$. Assume that $\psi \in X_{\mathcal{S}[x]}$. Since by definition ρ_s replaces each occurrence of $Q_\psi(x)$ by the conjunction over all $Q_\phi(x)$ such that $\rho_s(\phi) = \rho_r^{-1}(\phi) = \psi$, the previous equivalence implies that ρ_s maps $Q_\psi(x)$ to $Q_{\rho_r(\psi)}(x)$. Hence $\sigma_{\rho_r(\mathcal{S})} \circ \rho_s$ replaces Q_ψ with $\rho_r(\psi)$ and since $\sigma_{\mathcal{S}}$ replaces Q_ψ with ψ, $\rho_r \circ \sigma_{\mathcal{S}}$ coincides with $\sigma_{\rho_r(\mathcal{S})} \circ \rho_s$ for every formula in $\mathcal{R}$. ∎

7.2.4 Proposition

Let Γ and Λ be logic classes, then $\Gamma \circ \Lambda$ is also a logic class.

Proof. Let A be a finite alphabet. That $(\Gamma \circ \Lambda)_A$ is a Boolean algebra follows directly from the fact that for any two finite subalgebras $\mathcal{S} \subseteq \Lambda_A$ and $\mathcal{R} \subseteq \Gamma_{X_{\mathcal{S}[x]}}$, $\mathcal{R} \circ \mathcal{S}$ is again a Boolean algebra.

It remains to be shown that $\Gamma \circ \Lambda$ is closed under letter substitution. Let B be a second finite alphabet and $r\colon A \to B$ a map. Moreover, let $\phi \in (\Gamma \circ \Lambda)_B$, that is $\phi \in \mathcal{R} \circ \mathcal{S}$ for some finite Boolean subalgebras $\mathcal{S} \subseteq \Lambda_B$ and $\mathcal{R} \subseteq \Gamma_{X_{\mathcal{S}[x]}}$. We distinguish two cases:

In the first case, ϕ is a sentence in $\mathcal{S}$. Then, since $\mathcal{S} \subseteq \Lambda_B$ and Λ_B is closed under letter substitutions, it follows that $\rho_r(\phi)$ is a sentence in Λ_A and thus contained in $(\Gamma \circ \Lambda)_A$.

In the second case, ϕ is of the form $\sigma_{\mathcal{S}}(\psi)$ for some $\psi \in \mathcal{R}$. Then, by Lemma 7.2.3, there exists a letter substitution $s\colon X_{\rho_r(\mathcal{S})[x]} \to X_{\mathcal{S}[x]}$ such that

$$\rho_r(\sigma_{\mathcal{S}}(\psi)) = \rho_r(\phi) = \sigma_{\rho_r(\mathcal{S})}(\rho_s(\psi)).$$

Since Γ is a logic class, it is closed under letter substitution and hence $\rho_s(\psi) \in \Gamma_{X_{\rho_r(\mathcal{S})[x]}}$. Also $\rho_r(\mathcal{S})$ is a finite Boolean subalgebra of Λ_A. It follows immediately from the definition of $\Gamma \circ \Lambda$, that $\rho_r(\phi)$ is contained in $(\Gamma \circ \Lambda)_A$.

Hence, $\Gamma \circ \Lambda$ is a logic class. ∎

We are now ready to state the block product principle for *lp*-streams of typed stamps. We define the block product of two *lp*-streams of typed stamps as follows: Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps. Then $\mathbf{V}\Box\mathbf{W}$ is the *lp*-stream of typed stamps generated by all typed stamps $\mathsf{R}\Box\mathsf{S}$, where $\mathsf{S} \in \mathbf{W}(A \times 2^{\{x\}})$ and $\mathsf{R} \in \mathbf{V}(X_{\mathsf{S}[x]})$ and A is some finite alphabet.

Moreover let $\mathcal{V}$ and $\mathcal{W}$ be the *lp*-streams of languages corresponding to $\mathbf{V}$ and respectively $\mathbf{W}$. Then we denote by $\mathcal{V}\Box\mathcal{W}$ the *lp*-stream of languages corresponding to $\mathbf{V}\Box\mathbf{W}$.

7.2.5 Proposition

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps. Then for each alphabet A, the languages in $\mathcal{V}\Box\mathcal{W}(A)$ are Boolean combinations of languages recognised by typed stamps of the form $\mathsf{R}\Box\mathsf{S}$, where $\mathsf{R} \in \mathbf{V}(X_{\mathsf{S}[x]})$ and $\mathsf{S} \in \mathbf{W}(A \times 2^{\{x\}})$.

Proof. We need to show that by generating an *lp*-stream out of all typed stamps $\mathsf{R}\Box\mathsf{S}$, that is closing under division, trivial extension, finite direct products and *lp*-transformations, we only generate typed stamps that recognise Boolean combinations of the languages recognised by the block products.

Indeed, if $\varphi\colon A^* \to B^*$ is an *lp*-morphism, that is, it is a morphism generated by a function $r\colon A \to B$ and S is a typed stamp over $A \times 2^{\{x\}}$ and R over $X_{\mathsf{S}[x]}$, then $(\mathsf{R}\Box\mathsf{S}) \cdot \varphi$ is isomorphic to the block product of a typed stamp $\mathsf{T} \in \mathbf{W}(B \times 2^{\{x\}})$ and a typed stamp in $\mathbf{V}(X_{\mathsf{T}[x]})$. We omit the details, since the techniques are essentially the same as in the proof of Lemma 7.2.3.

Thus, *lp*-transformations do not contribute additional languages, nor do division nor trivial extension. The Boolean combinations then are recognised by finite direct products, which proves the claim. ∎

7.2.6 Theorem

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps. Then for each alphabet A, the languages in $\mathcal{V}\Box\mathcal{W}(A)$ are Boolean combinations of languages in $\mathcal{W}(A)$ and languages of the form $\tau_{\mathsf{S}}^{-1}(L)$, where τ_{S} is a transduction defined by some $\mathsf{S} \in \mathbf{W}(A \times 2^{\{x\}})$ and $L \in \mathcal{V}(X_{\mathsf{S}[x]})$.

The proof follows directly from the previous Proposition and Proposition 7.1.11.

We conclude this section by stating a connection between the block product of *lp*-streams of typed stamps and the substitution product of logic classes.

7.2.7 Corollary

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps and Γ and Λ be logic classes such that the corresponding *lp*-streams satisfy $L(\mathbf{V}) = L(\Gamma)$ and $L(\mathbf{W}) = L(\Lambda)$ then

$$L(\mathbf{V}\Box\mathbf{W}) = L(\Gamma \circ \Lambda)$$

7.3 The Block Product on Dense Stamps

In the previous sections, we were able to define the block product for typed stamps $\mathsf{R} = (X_{\mathsf{S}[x]}, \mu, M, p, X)$ and $\mathsf{S} = (A \times 2^{\{x\}}, \nu, N, q, Y)$. In the definition, it is essential that the Stone space of $\mathsf{S}[x]$, $X_{\mathsf{S}[x]}$ as a subset of Y is still finite and thus may be used as the alphabet for R.

The goal of this section is to characterise the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp. It is tempting to hope that if $\mathbf{V}$ and $\mathbf{W}$ are lp-streams of typed stamps, then the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp over some finite alphabet A should be isomorphic to a block product of the dense pro-$\mathbf{V}$ stamp and the dense pro-$\mathbf{W}$ stamp for appropriate alphabets.

We are going to prove a slightly weaker statement, that is: There exists a block product for dense stamps such that the block product of the dense pro-$\mathbf{V}$ and the dense pro-$\mathbf{W}$ stamp is a trivial extension of the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp. This weaker statement is not entirely surprising, since the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp is a syntactic object, whereas the block product rarely gives a syntactic object, already in the case of monoids.

To define said block-product on dense stamps, what seems a reasonable approach is to consider the dense pro-$\mathbf{W}$ stamp $\mathsf{R} := \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and the dense pro-$\mathbf{V}$ stamp over $X_{\mathsf{R}[x]}$ and to then use the current definition of block product. A problem that arises is that $X_{\mathsf{R}[x]}$ is in general not a finite alphabet.

However, as a Stone space, $X_{\mathsf{R}[x]}$ is profinite as a projective limit over all its finite continuous quotients (see for instance [Joh86]). We are thus going to introduce dense stamps over profinite alphabets in a coherent way, such that we may generalise the block product.

Dense Stamps over Profinite Alphabets

We say that an alphabet is profinite, if it is the projective limit of finite alphabets. Since each Stone space is a profinite set and finite sets may be seen as finite alphabets, each Stone space is a profinite alphabet (and vice versa).

A dense stamp over a profinite alphabet is a tuple (X, μ, M, p, Y), where X may be a profinite alphabet – all other requirements remain the same as for dense stamps, that is $\mu\colon X^* \to M$ is a morphism and so on.

We now prove that for each profinite alphabet X and each $\mathcal{C}$-variety of typed stamps $\mathbf{V}$, we may derive the notion of dense pro-$\mathbf{V}$ stamp over the profinite alphabet X from all dense pro-$\mathbf{V}$ stamps over finite quotients of X.

Let X be a profinite alphabet and let

$$\mathrm{Fin}(X) = \{A \mid A \text{ is a finite continuous quotient of } X\}.$$

be the set of continuous finite quotients of X. Moreover, let I be the index set of all projections $\pi_i\colon X \to A_i$, where $A_i \in \mathrm{Fin}(X)$ for $i \in I$. Then we may reconstruct the projective limit system underlying X in the following way.

7.3.1 Proposition

Let X be a profinite alphabet. The set $(A_i)_{i\in I}$, where $A_i \in \mathrm{Fin}(X)$ forms a projective limit system together with the connection maps $h_{i,j}\colon A_i \to A_j$ such that $h_{i,j}$ is a quotient map and commutes with the projections $\pi_j = h_{i,j} \circ \pi_i$. Its projective limit is X.

The proof is entirely straight-forward and thus omitted.

Observe that each of the quotient morphisms $h_{i,j}\colon A_i \to A_j$ induces a length preserving morphism $h_{i,j}^*\colon A_i^* \to A_j^*$. Let $\mathbf{V}$ be a $\mathcal{C}$-stream of typed stamps and let $\mathcal{V}$ be its corresponding stream of languages. Then, since $h_{i,j}$ is a quotient and $\mathcal{V}$ closed under inverse length-preserving morphisms, the map

$$\begin{aligned}(h_{i,j}^*)^{-1}\colon \mathcal{V}(A_j) &\to \mathcal{V}(A_i)\\ L &\mapsto (h_{i,j}^*)^{-1}(L)\end{aligned}$$

is well-defined and an embedding. Hence, its dual is a continuous quotient map $f_{i,j}\colon X_{\mathcal{V}(A_i)} \to X_{\mathcal{V}(A_j)}$ and the quotient morphism $\phi_{i,j}\colon M_{\mathcal{V}(A_i)} \to M_{\mathcal{V}(A_j)}$, where $M_{\mathcal{V}(A_i)}$ (respectively $M_{\mathcal{V}(A_j)}$) is the syntactic monoid of $\mathcal{V}(A_i)$ (respectively $\mathcal{V}(A_j)$) maps the equivalence class of $w \in A_i^*$, $[w]$ to $[h_{i,j}(w)]$. Together with Proposition 6.5.1 and Proposition 6.5.4, we obtain that the triple

$$\Phi_{i,j} = (h_{i,j}, \phi_{i,j}, f_{i,j})$$

is a morphism of dense stamps from $\widehat{\mathsf{R}}_{A_i}(\mathbf{V})$ to $\widehat{\mathsf{R}}_{A_j}(\mathbf{V})$. In particular, together with the morphisms $\Phi_{i,j}$, the family $(\widehat{\mathsf{R}}_{A_i}(\mathbf{V}))_{i\in I}$ contributes a projective system.

7.3.2 Proposition

Let X be a profinite alphabet and $\mathbf{V}$ a $\mathcal{C}$-stream of typed stamps. Then the projective limit of the system $(\widehat{\mathsf{R}}_{A_i}(\mathbf{V}))_{i\in I}$ exists.

Proof. Let $\widehat{\mathsf{R}}_{A_i}(\mathbf{V}) = (A_i, \mu_i, M_i, p_i, Y_i)$.

Since the morphisms $\Phi_{i,j} = (h_{i,j}, \phi_{i,j}, f_{i,j})$ contribute a projective system, also the monoids $(M_i)_{i\in I}$ with connection morphisms $\phi_{i,j}$ and the Stone spaces $(Y_i)_{i\in I}$ with connection morphisms $f_{i,j}$ are a projective system. The fact that both monoids and Stone spaces admit for projective limits in their category provides that $(M_i)_{i\in I}$ (respectively $(Y_i)_{i\in I}$) has a projective limit which we denote by M (respectively Y) and that there exists a map $p\colon M \to Y$ with dense image commuting with the projections in the sense that the diagram

$$\begin{array}{ccc} M & \xrightarrow{\;p\;} & Y \\ {\scriptstyle \pi_i^M}\downarrow & & \downarrow{\scriptstyle \pi_i^Y} \\ M_i & \xrightarrow[\;p_i\;]{} & Y_i \end{array}$$

commutes, where π_i^M and π_i^Y are the respective projections. In a similar fashion, we obtain that there exists a morphism $\mu\colon X^* \to M$ such that

$$\begin{array}{ccc} X^* & \xrightarrow{\mu} & M \\ {\scriptstyle \pi_i^*}\downarrow & & \downarrow{\scriptstyle \pi_i^M} \\ A_i^* & \xrightarrow[\mu_i]{} & M_i \end{array}$$

commutes. Here, π_i^* denotes as usual the extension of the map $\pi_i\colon X \to A_i$ to a monoid morphism of free monoids.

Hence, the projective limit of the system $(\widehat{\mathsf{R}}_{A_i}(\mathbf{V}))_{i\in I}$ is the tuple (X, μ, M, p, Y). ■

7.3.3 Definition

Let X be a profinite alphabet. The dense pro-$\mathbf{V}$ stamp over X is the projective limit of the system $(\widehat{\mathsf{R}}_{A_i}(\mathbf{V}))_{i\in I}$ and denoted by $\widehat{\mathsf{R}}_X(\mathbf{V})$. By $\mathcal{V}(X)$, we denote the set of all languages over X^* recognised by $\widehat{\mathsf{R}}_X(\mathbf{V})$.

In particular, we extract from the proof of Proposition 7.3.1, that in $\widehat{\mathsf{R}}_X(\mathbf{V}) = (X, \mu, M, p, Y)$, if $\widehat{\mathsf{R}}_{A_i}(\mathbf{V}) = (A_i, \mu_i, M_i, p_i, Y_i)$, then μ is the projective limit of the morphisms μ_i, M the projective limit of M_i and so forth.

The Block Product on Dense Stamps

We are now ready to state the block product for dense stamps using profinite alphabets and to characterise its relation to the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp.

7.3.4 Definition

Let A be a finite alphabet and let $\mathsf{S} = (A \times 2^{\{x\}}, \nu, N, q, Y)$ be a dense stamp. Moreover, let $\mathsf{R} = (X_{\mathsf{S}[x]}, \mu, M, p, X)$ be a dense stamp over the profinite alphabet $X_{\mathsf{S}[x]}$. The block product of R and S is the dense stamp

$$\mathsf{R}\Box\mathsf{S} = (A, \mu\Box\nu, M\Box N, p\Box q, X \times Y)$$

where

- $M\Box N$ is the block product of monoids
- $\mu\Box\nu$ is the morphism sending a to $(f_a, \nu(a))$, and
- $p\Box q$ sends the tuple (f, n) to $(p \circ f(1,1), q(n))$.

Comparing the definition to the block product for typed stamps shows that the two notions are compatible in the sense that if we consider a typed stamp as a (discrete)

dense stamp, then applying the block product for dense stamps yields the same result as the block product for typed stamps.

In the following, we consider *lp*-streams of typed stamps $\mathbf{V}$ and $\mathbf{W}$ and let $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$. As usual, we assume $\mathbf{V} \leftrightarrow \mathcal{V}$ and $\mathbf{W} \leftrightarrow \mathcal{W}$. In a similar fashion as for typed stamps, we derive a block product principle through the notion of transduction. Hence we define the transduction induced by W as the map

$$\begin{aligned}\tau_{\mathsf{W}} \colon A^* &\to X_{\mathsf{W}[x]}^* \\ w &\mapsto \iota_{\mathsf{W}}(w_{x=1}) \cdot \dots \cdot \iota_{\mathsf{W}}(w_{x=|w|})\end{aligned}$$

7.3.5 Proposition

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps, where $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$. Then each language recognised by the block product $\mathsf{V}\Box\mathsf{W}$ is a Boolean combination of languages in $\mathcal{W}(A)$ and languages of the form $\tau_{\mathsf{W}}^{-1}(L)$, where $L \in \mathcal{V}(X_{\mathsf{W}[x]})$.

Proof. The proof is essentially the same as the one of Proposition 7.1.11 and is thus presented in a condensed version. We denote by $i_A \colon A^* \to (A \times 2^{\{x\}})^*$ the inclusion. Let $\mathsf{V} = (X_{\mathsf{W}[x]}, \mu, M, p, X)$ and let $\mathsf{W} = ((A \times 2^{\{x\}}), \nu, N, q, Y)$. Since $\mathbf{W}$ is closed under *lp*-transformations, $\mathsf{W} \cdot i_A$ is an element of $\mathbf{W}$ and thus every language recognised by it is an element of $\mathcal{W}(A)$.

Recall that a language is recognised by $\mathsf{V}\Box\mathsf{W}$ if and only if it is of the form $\iota_{\mathsf{V}\Box\mathsf{W}}^{-1}(C)$, where C is a clopen of $X \times Y$. As a topological space $X \times Y$ is equipped with the product topology and the clopen sets are Boolean combinations of the sets $C_X \times Y$, where C_X is a clopen of X and $X \times C_Y$, where C_Y is a clopen of Y.

Hence, we assume without loss of generality that L is a language recognised by $\mathsf{V}\Box\mathsf{W}$ of the form

$$L = \iota_{\mathsf{V}\Box\mathsf{W}}^{-1}(C_X \times C_Y)$$

for some clopen $C_X \subseteq X$ and some clopen $C_Y \subseteq Y$. Letting $L_X = \iota_{\mathsf{V}}^{-1}(C_X)$ and $L_Y = \iota_{\mathsf{W}\cdot i_A}^{-1}(C_Y)$, we claim that $L = \tau_{\mathsf{W}}^{-1}(L_X) \cap L_Y$.

As in the proof of Proposition 7.1.11, we let $\mu\Box\nu(w) = (f_w, \nu(w))$ and obtain that $p \circ f_w(1,1) = \iota_{\mathsf{V}} \circ \tau_{\mathsf{W}}(w)$. Moreover $\iota_{\mathsf{V}\Box\mathsf{W}}(w) \in C_X \times C_Y$ if and only if $p \circ f_w(1,1) \in C_X$ and $\iota_{\mathsf{W}}(w) \in C_Y$. We can thus conclude for $w \in A^*$, that since $\iota_{\mathsf{W}} = \iota_{\mathsf{W}\cdot i_A}(w)$, the equality

$$\begin{aligned}w \in L &\Leftrightarrow \iota_{\mathsf{V}} \circ \tau_{\mathsf{W}}(w) \in C_X \text{ and } \iota_{\mathsf{W}}(w) \in C_Y \\ &\Leftrightarrow w \in \tau_{\mathsf{W}}^{-1}(L_X) \cap L_Y\end{aligned}$$

holds.

The converse direction follows along the same lines. ∎

The following two propositions exploit the fact that we are actually working on projective limits of dense stamps and those are uniquely determined by their projections.

7.3.6 Proposition

Let $\mathbf{W}$ be an *lp*-stream of typed stamps and let $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$. Then for each typed stamp $\mathsf{S} \in \mathbf{W}(A \times 2^{\{x\}})$ and language $L \subseteq X^*_{\mathsf{S}[x]}$, there exists a language $K \subseteq X^*_{\mathsf{W}[x]}$ such that

$$\tau_{\mathsf{S}}^{-1}(L) = \tau_{\mathsf{W}}^{-1}(K).$$

Proof. Let $\mathsf{W} = ((A \times 2^{\{x\}}), \mu, m, p, X)$ and $\mathsf{S} = ((A \times 2^{\{x\}}), \nu, N, q, Y)$. Observe that without loss of generality, we may assume that S is minimal with respect to trivial extension, since if S were a trivial extensions of some typed stamp R over $A \times 2^{\{x\}}$, then $X_{\mathsf{S}[x]}$ and $X_{\mathsf{R}[x]}$ are isomorphic.

It follows from the definition of W that S is a quotient of W. Let $\pi\colon X \to Y$ be the map defined by the projection morphism $\Pi\colon \mathsf{W} \to \mathsf{S}$. Then $\iota_{\mathsf{S}} = \pi \circ \iota_{\mathsf{W}}$. In particular, π restricts to a map $\pi_x\colon X_{\mathsf{W}[x]} \to X_{\mathsf{S}[x]}$ such that $\pi_x \circ \iota_{\mathsf{W}}$ restricted to $A^* \otimes \{x\}$ coincides with the restriction of ι_{S} to $A^* \otimes \{x\}$. Hence for $w \in A^*$ the equality

$$\iota_{\mathsf{S}}(w_{x=1}) \cdot \cdots \cdot \iota_{\mathsf{S}}(w_{x=|w|}) = \pi_x \circ \iota_{\mathsf{W}}(w_{x=1}) \cdot \cdots \cdot \pi_x \circ \iota_{\mathsf{W}}(w_{x=|w|})$$

holds and by extending π_x to a morphism $\pi_x^*\colon X^*_{\mathsf{W}[x]} \to X^*_{\mathsf{S}[x]}$

$$\tau_{\mathsf{S}}(w) = \pi_x^* \circ \tau_{\mathsf{W}}(w).$$

Hence for $L_S \subseteq X^*_{\mathsf{S}[x]}$, setting $K = (\pi_x^*)^{-1}(L)$ provides us with

$$\begin{aligned} w \in \tau_{\mathsf{S}}^{-1}(L) &\Leftrightarrow \tau_{\mathsf{S}}(w) \in L \\ &\Leftrightarrow \pi_x^* \circ \tau_{\mathsf{W}}(w) \in L \\ &\Leftrightarrow \tau_{\mathsf{W}}(w) \in (\pi_x^*)^{-1}(L) \\ &\Leftrightarrow w \in K \end{aligned}$$

and thus proves the claim. ∎

7.3.7 Proposition

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps, where $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$. Moreover let $\mathbf{V} \leftrightarrow \mathcal{V}$. Then for each language $L \in \mathcal{V}(X_{\mathsf{W}[x]})$, there exists a typed stamp $\mathsf{S} \in \mathbf{W}$ and a language $K \in \mathcal{V}(X_{\mathsf{S}[x]})$ such that

$$\tau_{\mathsf{W}}^{-1}(L) = \tau_{\mathsf{S}}^{-1}(K).$$

Proof. Let $\mathsf{W} = (A \times 2^{\{x\}}, \nu, N, q, Y)$ and $\mathsf{V} = (X_{\mathsf{W}[x]}, \mu, M, p, X)$. Since V is the projective limit of the system

$$\left\{ \widehat{\mathsf{R}}_{X_{\mathsf{R}[x]}}(\mathbf{V}) \mid \mathsf{R} \text{ is a finite quotient of } \mathsf{W} \right\},$$

for each $\mathsf{R} \in \mathbf{W}$, there exists a projection $\Pi^{\mathsf{R}}\colon \mathsf{V} \to \widehat{\mathsf{R}}_{X_{\mathsf{R}[x]}}(\mathbf{V})$, where we let $\Pi^{\mathsf{R}} = (r^{\mathsf{R}}, \phi^{\mathsf{R}}, \pi^{\mathsf{R}})$ and

$$\mathsf{V}^{\mathsf{R}} := \widehat{\mathsf{R}}_{X_{\mathsf{R}[x]}}(\mathbf{V}) = (X_{\mathsf{R}[x]}, \mu^{\mathsf{R}}, M^{\mathsf{R}}, p^{\mathsf{R}}, X^{\mathsf{R}}).$$

Summarising the situation in a diagram, we obtain that

$$\begin{array}{ccc} X^*_{\mathsf{W}[x]} & \xrightarrow{\iota_{\mathsf{V}}} & X \\ {\scriptstyle r^{\mathsf{R}}}\downarrow & & \downarrow{\scriptstyle \pi^{\mathsf{R}}} \\ X^*_{\mathsf{R}[x]} & \xrightarrow[\iota_{\mathsf{V}^{\mathsf{R}}}]{} & X^{\mathsf{R}} \end{array}$$

commutes.

Let $L \in \mathcal{V}(X_{\mathsf{W}[x]})$ be a language. Since $L \in \mathcal{V}(X_{\mathsf{W}[x]})$, there exists a clopen $C \subseteq X$ such that $L = \iota_{\mathsf{V}}^{-1}(C)$ and in particular, since C is clopen, there exists an $n \in \mathbb{N}$ and for each $i = 1, \ldots, n$, a typed stamp $\mathsf{S}_i \in \mathsf{W}$ and a set $C^{\mathsf{S}_i} \subseteq X^{\mathsf{S}_i}$ such that C is a (finite) Boolean combination of the sets $(\pi^{\mathsf{S}_i})^{-1}(C^{\mathsf{S}_i})$. Observe that this is true, since X itself is the projective limit of the spaces X^{R} for $\mathsf{R} \in \mathbf{W}$ and as such equipped with the product topology.

Let S be the product $\mathsf{S}_1 \times \cdots \times \mathsf{S}_n$. Then, for each $i = 1, \ldots, n$ there exist quotient morphisms $r_i^{\mathsf{S}} \colon X_{\mathsf{S}[x]} \to X_{\mathsf{S}_i[x]}$ and each r_i^{S} extends to a morphism from $X^*_{\mathsf{S}[x]}$ to $X^*_{\mathsf{S}_i[x]}$. We denote by $\pi_i^{\mathsf{S}} \colon X^{\mathsf{S}} \to X^{\mathsf{S}_i}$ the projections. Summarising again in a diagram, we have

$$\begin{array}{ccc} X^*_{\mathsf{W}[x]} & \xrightarrow{\iota_{\mathsf{V}}} & X \\ {\scriptstyle r^{\mathsf{S}}}\downarrow & & \downarrow{\scriptstyle \pi^{\mathsf{S}}} \\ X^*_{\mathsf{S}[x]} & \xrightarrow[\iota_{\mathsf{V}^{\mathsf{S}}}]{} & X^{\mathsf{S}} \\ {\scriptstyle r_i^{\mathsf{S}}}\downarrow & & \downarrow{\scriptstyle \pi_i^{\mathsf{S}}} \\ X^*_{\mathsf{S}_i[x]} & \xrightarrow[\iota_{\mathsf{V}^{\mathsf{S}_i}}]{} & X^{\mathsf{S}_i} \end{array}$$

Let

$$K = \iota_{\mathsf{V}^{\mathsf{S}}}^{-1}\left(\bigcap_{i=1}^{n} (\pi^{\mathsf{S}_i})^{-1}(C^{\mathsf{S}_i})\right).$$

Then $K \in \mathcal{V}(X_{\mathsf{S}[x]})$ and since the diagram above commutes, $L = (r^{\mathsf{S}})^{-1}(K)$. Recall that r^{S} is precisely the monoid morphism induced by the last component of the quotient morphism from W to S, which implies $r^{\mathsf{S}} \circ \iota_{\mathsf{W}} = \iota_{\mathsf{S}}$. We obtain

$$\begin{aligned} \tau_{\mathsf{W}}(w) \in L &\Leftrightarrow \iota_{\mathsf{W}}(w_{x=1}) \cdot \cdots \cdot \iota_{\mathsf{W}}(w_{x=|w|}) \in L \\ &\Leftrightarrow r^{\mathsf{S}}(\iota_{\mathsf{W}}(w_{x=1}) \cdot \cdots \cdot \iota_{\mathsf{W}}(w_{x=|w|})) \in K \\ &\Leftrightarrow r^{\mathsf{S}} \circ \iota_{\mathsf{W}}(w_{x=1}) \cdot \cdots \cdot r^{\mathsf{S}} \circ \iota_{\mathsf{W}}(w_{x=|w|}) \in K \\ &\Leftrightarrow \iota_{\mathsf{S}}(w_{x=1}) \cdot \cdots \cdot \iota_{\mathsf{S}}(w_{x=|w|}) \in K \\ &\Leftrightarrow \tau_{\mathsf{S}}(w) \in K \end{aligned}$$

which proves the claim. ∎

As a consequence of the previous observations, we now obtain a precise relationship, between the languages recognised by $\mathbf{V}\Box\mathbf{W}$ and those recognised by $\mathsf{V}\Box\mathsf{W}$.

7.3.8 Theorem

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps, where $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$. Then

$$\mathcal{V}\Box\mathcal{W}(A) = L(\mathsf{V}\Box\mathsf{W}).$$

Proof. By Proposition 7.3.5, each language recognised by $\mathsf{V}\Box\mathsf{W}$ is a Boolean combination of languages in $\mathcal{W}(A)$ and languages of the form $\tau_{\mathsf{W}}^{-1}(L)$, where L is recognised by V.

Hence, Proposition 7.3.6 implies that $\mathcal{V}\Box\mathcal{W}(A) \subseteq L(\mathsf{V}\Box\mathsf{W})$. The converse inclusion follows from Proposition 7.3.7. ∎

7.3.9 Theorem

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps, where $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$. Then the block product $\mathsf{V}\Box\mathsf{W}$ is a trivial extension of $\widehat{\mathsf{R}}_A(\mathbf{V}\Box\mathbf{W})$.

Proof. By Theorem 7.3.8, $\mathsf{V}\Box\mathsf{W}$ and $\widehat{\mathsf{R}}_A(\mathbf{V}\Box\mathbf{W})$ recognise the same languages, hence they have isomorphic Stone spaces. Since the monoid component of $\widehat{\mathsf{R}}_A(\mathbf{V}\Box\mathbf{W})$ is the syntactic monoid of $\mathcal{V}\Box\mathcal{W}(A)$, the claim follows. ∎

Finally, reformulating our observations again, such that they match the perspective from logic, we obtain as a Corollary from Theorem 7.3.8.

7.3.10 Corollary

Let $\mathbf{V}$ and $\mathbf{W}$ be *lp*-streams of typed stamps, where $\mathsf{W} = \widehat{\mathsf{R}}_{A\times 2^{\{x\}}}(\mathbf{W})$ and $\mathsf{V} = \widehat{\mathsf{R}}_{X_{\mathsf{W}[x]}}(\mathbf{V})$ and let Γ and Λ be logic classes such that $L(\mathbf{V}) = L(\Gamma)$ and $L(\mathbf{W}) = L(\Lambda)$. Then

$$L(\Gamma \circ \Lambda) = L(\mathsf{V}\Box\mathsf{W}).$$

7.4 Summary

We used substitution and transduction, which in the first place work purely on logical formulae and defined a notion of transduction on typed stamps, such that the transduction defined by the (finitely many) formulae coincided with the transduction of the typed stamp, if both recognise the same languages (Corollary 7.1.9). We then defined the block product for typed stamps (Definition 7.1.10) and, using the previous relation, showed that the block product and substitution stand in relation (Corollary 7.1.12).

Leaving the local world, where we worked on single typed stamps, we introduced the notion of logic class, in order to be able to set $\mathcal{C}$-streams of typed stamps and Boolean algebras of logical formulae in relation. That generalisation also served to characterise the languages recognised by the block product of two $\mathcal{C}$-streams of typed stamps, which again is a $\mathcal{C}$-stream of typed stamps, in terms of logic (Corollary 7.2.7).

The final goal was then to characterise, for two $\mathcal{C}$-streams of typed stamps $\mathbf{V}$ and $\mathbf{W}$, the dense pro $\mathbf{V}\Box\mathbf{W}$-stamp of the block product. This was done in terms of a generalisation of the block product of typed stamps to dense stamps. However, it had to be adjusted to also work for profinite alphabets, which we unavoidably encounter, if we want to define an analogue of substitution and transduction for possibly infinite Boolean algebras of formulae.

Finally, we were able to show that the block product of the dense pro-$\mathbf{V}$ and the dense pro-$\mathbf{W}$ stamp recognises the same languages as the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp and thus gave a more constructive characterisation of the dense pro-$\mathbf{V}\Box\mathbf{W}$ stamp.

7.5 Further Research

An instance on which the now established theory could be tried out was considered in [BKR09], where it was possible to show that majority logic with the $<$-predicate and two variables ($\mathrm{Maj}_2[<]$) is not capable of recognising non-solvable group languages, as for instance A_5. The arguments presented in that paper seem to be of topological nature, since the authors consider sequences of words, where it is important that the length of the words grows with the index. More precisely the elements of these sequences at index n give rise to a pair of words which cannot be distinguished by any formula in said fragment of logic with at most n majority quantifiers. We aim at a topological reproof, using the block product, deriving equations inductively for quantifier depth $n+1$ in dependence of the equations for depth n. Although there are equations for the regular languages inside $\mathrm{Maj}_2[<]$, since they form a pseudovariety, it may still be necessary to consider ultrafilter equations in the process of the induction.

But already equations for $\mathrm{Maj}[<]$ of depth 2, which coincides with $\mathrm{Maj}_2[<]$ of depth two, could be interesting on their own, to consider a reproof of the separation of $\mathrm{Maj}[<]$ depth 3 by equations.

8

Conclusion

We investigated the ties between algebra and topology outside of the regular languages, dividing the work in two forks: The investigation of visibly pushdown languages and of classes which are related to logic.

For the visibly pushdown languages, we were able to prove that they admit for finite algebraic recognising objects – Ext-algebras. Here, the additional algebraic structure played a key role in defining finite structures for these non-regular languages. These prepared the ground for the construction of the topological perspective for the visibly pushdown languages. Quite similarly to the regular languages, it was possible to construct a topological space via the completion of a metric space, which turned out to be the free profinite Ext-algebra and the Stone space of the visibly pushdown languages. In [GGP10], it was shown that the Stone space of a Boolean algebra of languages (relative to A^*) is a monoid, if and only if the Boolean algebra consists of regular languages. Here, we show that if a Boolean algebra of languages (now relative to $A^{\triangle}$) is a subset of all VPLs, then its Stone space is an Ext-algebra. In order to prove the converse direction – in the style of [GGP10] – one would have to consider a notion of recogniser for arbitrary Boolean algebras relative to $A^{\triangle}$ – choosing the Stone space should, to the best of the authors knowledge, then result in "The Stone space of a Boolean algebra relative to $A^{\triangle}$ is an Ext-algebra if and only if the Boolean algebra consists only of VPL".

The developed algebraic and topological notions contributed to form an Eilenberg-Reiterman-like relationship in the sense that each pseudo-variety of Ext-algebras corresponds uniquely to a pseudo-variety of VPL and both of these are uniquely determined by a set of equations over the free profinite Ext-algebra. These equations now differ from the well-known equations over profinite words (on $\widehat{A^*}$) in the sense, that they also admit for so-to-say profinite operations, such as $\mathrm{ext}^{\omega}_{u,v}$, which is a map from $\widehat{A^{\triangle}}$ to $\widehat{A^{\triangle}}$.

To employ these algebraic and topological findings, we subsequently chose a subclass of the visibly pushdown languages for examination: The class $\mathcal{M}$Ext consisting

of all languages that are the intersection of a regular language with the set of all well-matched words. The decidability whether a VPL belongs to $\mathcal{M}\text{Ext}$ was part of the work in [BLS06]. After showing that this class forms a pseudo-variety and hence has a defining set of equations, we gave sound equations for $\mathcal{M}\text{Ext}$. Through these equations, it was possible to prove in a rather straight-forward way, for some languages, that they do not belong to $\mathcal{M}\text{Ext}$. In particular, we were able to show that one of the equations is sound for VCL and conjecture that it might already be complete for it.

Evidently, for the case of VPL, algebra and topology are quite as neatly tied as in the case of the regular languages.

The second fork of the work was concerned with more general concepts of algebraic recognisers for non-regular language classes and their ties to topology. The introduced recognisers – typed stamps – were specifically motivated by classes of languages definable in fragments of logic. Hence, we included the morphism in typed stamps to be able to capture the peculiarities of for instance, the quasi-aperiodic languages and other classes mainly defined by logic. Typed stamps were derived from typed monoids [KLR05].

Due to their additional set-component, typed stamps are rather adapt to recognise (finite) Boolean algebras instead of languages. While the first few section were purely algebraical, already at the introduction of syntactic typed stamps, it became evident, that these are strongly tied to discrete topological objects, since the set-component of the syntactic typed stamp of a Boolean algebra $\mathcal{B}$ had the Stone space of $\mathcal{B}$ as set-component. Staying on the algebraic side, it was possible to derive an Eilenberg-like theorem for $\mathcal{C}$-streams of typed stamps and $\mathcal{C}$-streams of languages. These $\mathcal{C}$-streams of typed stamps come with the advantage that any set of typed stamps may generate a $\mathcal{C}$-stream but with the disadvantage that some of the typed stamps have – through trivial extension – incredibly large monoid components. In order to be able to build a meaningful notion of topological objects corresponding to a $\mathcal{C}$-stream it was hence necessary to restrict to slightly smaller classes, so-called inherent classes. These also come with a one-to-one correspondence between $\mathcal{C}$-streams of typed stamps and languages, but do not contain trivial extensions. However, we cannot say that any set of typed stamps (explicitly) generates such an inherent class.

The superordinate goal was to build one topological object from a class of typed stamps, which we can in later chapters use, to define a block product on, much like in [AW95]. Via a projective limit system over an inherent class of a $\mathcal{C}$-stream $\mathbf{V}$, it was possible to construct such a topological object – the dense pro-$\mathbf{V}$ stamp, now in the category of dense stamps, in which typed stamps may be seen as the discrete topological objects. It was then possible to show that the dense pro-$\mathbf{V}$ stamp also has a characterisation purely on the language side – a characterisation which makes the objects slightly more accessible and helps in grasping the connections to languages recognised by block products in later chapters. Finally, we were able to prove a Reiterman-like theorem, completing the triad between algebra, languages and topology and calculated two easy examples.

Having established all the necessary tools to tackle the block product topologically

in the fashion of [AW95], we first uncoiled the relations between substitution, transductions and typed stamps and defined a block product for typed stamps mirroring substitution on the side of logic. We then gradually increased complexity, first raising substitution to classes of formulae and the block product to $\mathcal{C}$-streams of typed stamps, to finally raise the block product to dense stamps.

A necessary complication there was to introduce profinite alphabets – which already was done similarly for the regular case in [AW95]. Using profinite alphabets, a block product principle connecting the block product for dense stamps and profinite transductions made it possible to describe the links between classes definable by logic and the block product of the respective dense stamps of these classes.

Further Research

At the end of each chapter, we gave directions for possible further research – we refer to those for more concrete problems. Here, we give a broader and more long-term directed overview.

Decidability for Visibly Pushdown Languages by Equations?

Recall that a VPL L satisfies an equation of profinite well-matched words $[u \leftrightarrow v]$, if the unique uniformly continuous extension of a morphism φ into an Ext-algebra recognising L maps both words u, v to the same element, that is $\widehat{\varphi}(u) = \widehat{\varphi}(v)$. The decisive advantage of having a notion of equation that has an interpretation on finite recognising objects is that in many cases, whether $[u \leftrightarrow v]$ holds can effectively computed. In fact, if u is the limit of the sequence of well-matched words u_n and v of v_n, then the sequence $\varphi(u_n)$ becomes stationary and equal to $\widehat{\varphi}(u)$ after finitely many steps.

Hence there is an algorithm deciding whether an equation holds for any VPL, whenever the profinite well-matched words in the equation are constructively given as the limits of sequences. This is, for instance, the case for $\mathrm{ext}^{\omega}_{ab}(x)$, which is the limit of $\mathrm{ext}^{n!}_{a,b}(x)$. More concretely, if the VPL is given by a VPA, we may convert it to an Ext-algebra via the procedure given in Proposition 5.2.13. On this Ext-algebra it is possible to evaluate the equation as described above.

Evidently, since the free profinite Ext-algebra contains uncountably many elements, not all converging sequences are finitely representable and consequently having a complete and sound set of equations does not automatically imply decidability. This raises a cascade of questions:
Which subclasses of VPL admit for a description through such well-behaving equations (for which the algorithm above is applicable)? For instance, in the regular languages, most known classes are expressible through ω-terms, which do imply decidability by a similar argument over monoids. Does it hold for VPL, that most naturally emerging classes are representable by equations over $\mathrm{ext}^{\omega}_{ab}$-terms or is it already for apparently simple classes the case that in terms of equations, more complex constructs are

needed? If so - how are these represented and do they still imply decidability, as they are limits of finitely representable sequences?

These questions are still relatively vague. More concretely, we already mentioned that examining VPL contained in complexity classes could provide interesting results. This is particularly interesting when talking about decidability: Taking a step back to results achieved for regular languages, it was shown in [GKP16], for a fragment of logic, that by finding equations for the fragment of logic, equations for the regular languages therein may be obtained, which then imply decidability. This naturally raises the question, whether the same procedure is possible for VPL. This might prove a tad more difficult than in [GKP16], since these result intrinsically rely on the fact that the free profinite monoid is a quotient of the Stone-Čech compactification, over which the equations for the fragment of logic were formed. Thus, it is possible to obtain equations for regular languages via projection. The free profinite Ext-algebra is *not* a quotient of the Stone-Čech compactification and hence this technique is in general not possible. However, since the free Ext-algebra $A^{\triangle}$ may be understood as a subset of the free monoid A^*, this gives rise to the assumption that we may interpret $\widehat{A^{\triangle}}$ as a subset of $\beta(A^*)$ and that equations for circuit classes (or fragments of logic) might via some sort of transitive closure over those equations lead to equations for VPL nonetheless.

Equations for Circuits Through Typed Stamps

Equations are a ray of hope for proving separations, since in theory it suffices to find just one equation that is violated by some language in order to prove a separation. For instance, it would not even be necessary to find a complete set of equations for $\mathbf{AC}^0$ – just one equation that holds for it but is violated by Parity would be enough to reprove [FSS84]. Consequently the hard part is *finding* the equations. In general, they are highly non-constructive. Here, we hope to be able to use an inductive approach for the construction of equations using typed stamps and the block product: First, consider fragments of logic (resp. circuit classes) with constant and small quantifier depth, for instance depth one or two, and find equations for them. Then use the block product and in particular the information we gained on the Stone space of the block product, to derive equations for higher depth fragments.

Already the first step, in which we wish to examine classes of considerably small depth, has interesting problems on its own, for instance it is unknown whether $\mathbf{CC}^0$ can compute 1^* and it is also unknown whether it is equal to $\mathbf{NP}$. Quite similarly, $\mathbf{TC}^0$ depth 3 and 4 are not yet separated. Here, calculating the dense pro-$\mathbf{V}$ stamps (where $\mathbf{V}$ is the respective class) and using them to derive equations should be of its own interest.

For the inductive step, one could then use the block product for dense stamps to obtain equations for larger classes. We mentioned previously that majority logic with two variables and the $<$-predicate is a strong candidate to reprove the results of [BKR09] as a sanity check for the framework. An obvious second candidate, which should however prove much harder, is $\mathbf{FO}[\mathcal{N}]$ (or equivalently $\mathbf{AC}^0$). A non-

probabilistic reproof [FSS84] would be a major breakthrough for the topological approach. However, it is to be suspected that to obtain precisely this, a lot of preprocessing is necessary. For instance in the regular case, Almeida and Weil [AW98] were able to derive equations for block products of pseudo-varieties of finite monoids, using a similar strategy going over profinite alphabets and - in addition - the derived category theorem [Til87] (a one-sided version of the Kernel theorem by Rohdes and Tilson [RT89]). It might thus be a wise choice to first establish a kernel-theorem for typed stamps and use this result and the results of the previous chapter in the fashion of Almeida and Weil, to obtain equations for large logic classes.

As a slight simplification, one could consider weak block-products instead of block products in full generality. Weak block products were, for instance, considered in [BKM13], where it was proven that iterated weak block products have connections to linear-size circuit classes. An advantage is that for an iterated weak block product, in contrast to iterated block products in general, the transduction for each iteration step is the same. Hence, in examining the transduction, on could probably derive equations in a much more constructive fashion for linear-size circuit classes. To then draw conclusions for polynomial-size circuits, padding (e.g. artificially transforming a polynomial-size circuit to a linear size one by increasing the input size with sufficiently many 1's) could be of use – one could show that the padded language violates the equation for linear size circuits and hence the original language cannot be accepted by circuits of polynomial size. This approach particularly lives of having equations for non-regular languages, since padded languages are in general not regular.

Index

Symbols

A

B

C

D

Bibliography

[AKMV05] Rajeev Alur, Viraj Kumar, Parthasarathy Madhusudan, and Mahesh Viswanathan. Congruences for visibly pushdown languages. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 1102–1114, 2005.

[Ali16] Saeid Alirezazadeh. On pseudovarieties of forest algebras. *Int. J. Found. Comput. Sci.*, 27(8):909–942, 2016.

[Alm95] Jorge Almeida. *Finite Semigroups and Universal Algebra*. World Scientific Publishing Co. Inc., Singapore, 1995.

[AM04] Rajeev Alur and Parthasarathy Madhusudan. Visibly pushdown languages. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 202–211, 2004.

[AW95] Jorge Almeida and Pascal Weil. Free profinite semigroups over semidirect products. *Russian Mathem.*, 39:1–28, 1995.

[AW98] Jorge Almeida and Pascal Weil. Profinite categories and semidirect products. *Journal of Pure and Applied Algebra*, 123:1–50, 1998.

[BCGK17] Célia Borlido, Silke Czarnetzki, Mai Gehrke, and Andreas Krebs. Stone duality and the substitution principle. In *26th EACSL Annual Conference on Computer Science Logic, CSL 2017, August 20-24, 2017, Stockholm, Sweden*, pages 13:1–13:20, 2017.

[BCST92] David A. Mix Barrington, Kevin J. Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *J. Comput. Syst. Sci.*, 44(3):478–499, 1992.

[Bir35] Garrett Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[BIS88] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. In *Proceedings: Third Annual Structure in Complexity Theory Conference, Georgetown University, Washington, D. C., USA, June 14-17, 1988*, pages 47–59, 1988.

[BKM13] Christoph Behle, Andreas Krebs, and Mark Mercer. Linear circuits, two-variable logic and weakly blocked monoids. *Theor. Comput. Sci.*, 501:20–33, 2013.

[BKR09] Christoph Behle, Andreas Krebs, and Stephanie Reifferscheid. Regular languages definable by majority quantifiers with two variables. In *Developments in Language Theory, 13th International Conference, DLT 2009, Stuttgart, Germany, June 30 - July 3, 2009. Proceedings*, pages 91–102, 2009.

[BLS06] Vince Bárány, Christof Löding, and Olivier Serre. Regularity problems for visibly pushdown languages. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, pages 420–431, 2006.

[Boj15] Mikołaj Bojańczyk. Recognisable languages over monads. *CoRR*, abs/1502.04898, 2015.

[BS73] Janusz A. Brzozowski and Imre Simon. Characterizations of locally testable events. *Discrete Mathematics*, 4(3):243–271, 1973.

[BT88] David A. Mix Barrington and Denis Thérien. Finite monoids and the fine structure of NC^1. *J. ACM*, 35(4):941–952, 1988.

[Büc60] Julius Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(16):66–92, 1960.

[BW08] Mikołaj Bojańczyk and Igor Walukiewicz. Forest algebras. In *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas].*, pages 107–132, 2008.

[CK16] Silke Czarnetzki and Andreas Krebs. Using duality in circuit complexity. In *Language and Automata Theory and Applications - 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings*, pages 283–294, 2016.

[CKL18] Silke Czarnetzki, Andreas Krebs, and Klaus-Jörn Lange. Visibly pushdown languages and free profinite monoids. *CoRR*, abs/1810.12731, 2018.

[Eil76] Samuel Eilenberg. *Automata, languages, and machines. Vol. B.* Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976.

[ES52] Samuel Eilenberg and Norman Earl Steenrod. *Foundations of Algebraic Topology.* Princeton University Press, 1952.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[Geh11] Mai Gehrke. Duality and recognition. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, pages 3–18, 2011.

[GGP08] Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin. Duality and equational theory of regular languages. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings*, pages 246–257, 2008.

[GGP10] Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin. A topological approach to recognition. In *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, pages 151–162. Springer Berlin Heidelberg, 2010.

[GKP16] Mai Gehrke, Andreas Krebs, and Jean-Éric Pin. Ultrafilters on words for a fragment of logic. *Theor. Comput. Sci.*, 610:37–58, 2016.

[GL84] Yuri Gurevich and Harry R. Lewis. A logic for constant-depth circuits. *Information and Control*, 61(1):65–74, 1984.

[GPR16] Mai Gehrke, Daniela Petrisan, and Luca Reggio. The schützenberger product for syntactic spaces. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 112:1–112:14, 2016.

[GPR17] Mai Gehrke, Daniela Petrisan, and Luca Reggio. Quantifiers on languages and codensity monads. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017.

[Hah15] Michael Hahn. An algebraic perspective on visibly pushdown languages. Master's thesis at the University of Tübingen, 2015.

[Imm89] Neil Immerman. Expressibility and parallel complexity. *SIAM Journal on Computing*, 18:625–638, 1989.

[Joh86] Peter T. Johnstone. *Stone Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1986.

[Kle56] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. In *Automata studies*, pages 3–41. Princeton University Press, Princeton, N. J., 1956.

[KLR05] Andreas Krebs, Klaus-Jörn Lange, and Stephanie Reifferscheid. Characterizing TC^0 in terms of infinite groups. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 496–507, 2005.

[KR65] Kenneth Krohn and John Rhodes. The Algebraic Theory of Machines I. *Transactions of the American Mathematical Society*, 116:450–464, 1965.

[Lan04] Klaus-Jörn Lange. Some results on majority quantifiers over words. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 123–129, 2004.

[Lud18] Michael Ludwig. Tree-structured problems and parallel computation. PhD thesis at the University of Tübingen, 2018.

[MP71] Robert McNaughton and Seymour Papert. *Counter-free automata*. M.I.T. Press Cambridge, Mass, 1971.

[Pin12] Jean-Éric Pin. Equational descriptions of languages. *Int. J. Found. Comput. Sci.*, 23(6):1227–1240, 2012.

[Pin16] Jean-Éric Pin. Mathematical foundations of automata theory, 2016.

[Pip97] Nicholas Pippenger. Regular languages and stone duality. *Theory Comput. Syst.*, 30(2):121–134, 1997.

[PS05] Jean-Éric Pin and Howard Straubing. Some results on $\mathcal{C}$-varieties. *ITA*, 39(1):239–262, 2005.

[Rei82] Jan Reiterman. The Birkhoff theorem for finite algebras. *Algebra Universalis*, 14:1–10, 1982.

[RS59] Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[RS97] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*. Springer-Verlag, Berlin, Heidelberg, 1997.

[RT89] John Rhodes and Bret Tilson. The kernel of monoid morphisms. *Journal of Pure and Applied Algebra*, 62(3):227 – 268, 1989.

[Sak76] Jacques Sakarovitch. An algebraic framework for the study of the syntactic monoids application to the group languages. In *Mathematical Foundations of Computer Science 1976, 5th Symposium, Gdansk, Poland, September 6-10, 1976, Proceedings*, pages 510–516, 1976.

[Sch56] Marcel Paul Schützenberger. Une théorie algébrique du codage. *Séminaire Dubreil. Algèbre et théorie des nombres*, 9:1–24, 1955-1956.

[Sch64] Marcel Paul Schützenberger. On the synchronizing properties of certain prefix codes. *Information and Control*, 7(1):23–36, 1964.

[Sto36] Marshall H. Stone. The theory of representations for boolean algebras. *Transactions of the American Mathematical Society*, 40(1):37–111, 1936.

[Str94] Howard Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, Basel, Switzerland, 1994.

[Str02] Howard Straubing. On logical descriptions of regular languages. In *LATIN 2002: Theoretical Informatics, 5th Latin American Symposium, Cancun, Mexico, April 3-6, 2002, Proceedings*, pages 528–538, 2002.

[Til87] Bret Tilson. Categories as algebra: An essential ingredient in the theory of monoids. *Journal of Pure and Applied Algebra*, 48(1):83 – 198, 1987.

[TT07] Pascal Tesson and Denis Thérien. Logic meets algebra: the case of regular languages. *Logical Methods in Computer Science*, 3(1), 2007.

[UACM16] Henning Urbat, Jirí Adámek, Liang-Ting Chen, and Stefan Milius. One eilenberg theorem to rule them all. *CoRR*, abs/1602.05831, 2016.

[UACM17] Henning Urbat, Jirí Adámek, Liang-Ting Chen, and Stefan Milius. Eilenberg theorems for free. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 43:1–43:15, 2017.